Lesson2:
Modelling the Web with Simple Statistical Descriptive Text Models
Unit1:
Counting Words and Documents

Rene Pickhardt

Introduction to Web Science Part 2

Emerging Web Properties

WeST
People and Knowledge Networks

Institute for Web Science and Technologies · University of Koblenz-Landau, Germany

Completing this unit you should

- Understand why we selected simple English Wikipedia as a toy example for modeling the web
- Understand that a task already as simple as counting words includes modeling choices
- Be familiar with the term “unique word token”
- Know some basic tools to count words and documents

Our toy example for the Web

- Simple English Wikipedia

- That is a strong assumption (e.g. Wikipedia has almost no dead links)

- Pro side:
  - It is available
  - It fits in memory / on disk
  - Calculations won't take too much time
  - Already big enough

A question of size

- Let us count documents

$ wc -l simple-20160801-1-article-per-line

119753 simple-20160801-1-article-per-line

- About 120k articles in simple English Wiki

Number of words depends heavily on the way we count and model what a word is

- 16,491,538 words with
  - re.findall(ur'\w+', text)
- 15,916,471 words with
  - text.split()
- 15,807,612 words with
  - text.split(" ")
- 15,685,177 words
  - with re.findall('[a-z]+', text.tolower())

Results are even skewer when we look at unique word tokens

- 292,932 words with
  - re.findall(ur'\w+', text)
- 842,272 words with
  - text.split()
- 909,777 words with
  - text.split(" ")
- 225,192 words with
  - re.findall('[a-z]+', text.tolower())

What is a unique word token?

- “This is an example of an interesting text”
- 8 words
- The token “an” occurs twice
  - Only 7 unique word tokens

Lesson2:
Modelling the Web with Simple Statistical Descriptive Text Models
Unit2:
Typical size of a document

Rene Pickhardt

Introduction to Web Science Part 2

Emerging Web Properties

WeST
People and Knowledge Networks

Completing this unit you should

- Be familiar with some basic statistical objects like
  - Median
  - Mean
  - Histograms

- Should be able to relate a histogram to its cumulative distribution function

What is the typical length of a document?

- We saw
  - 16491538 words
  - 119754 documents

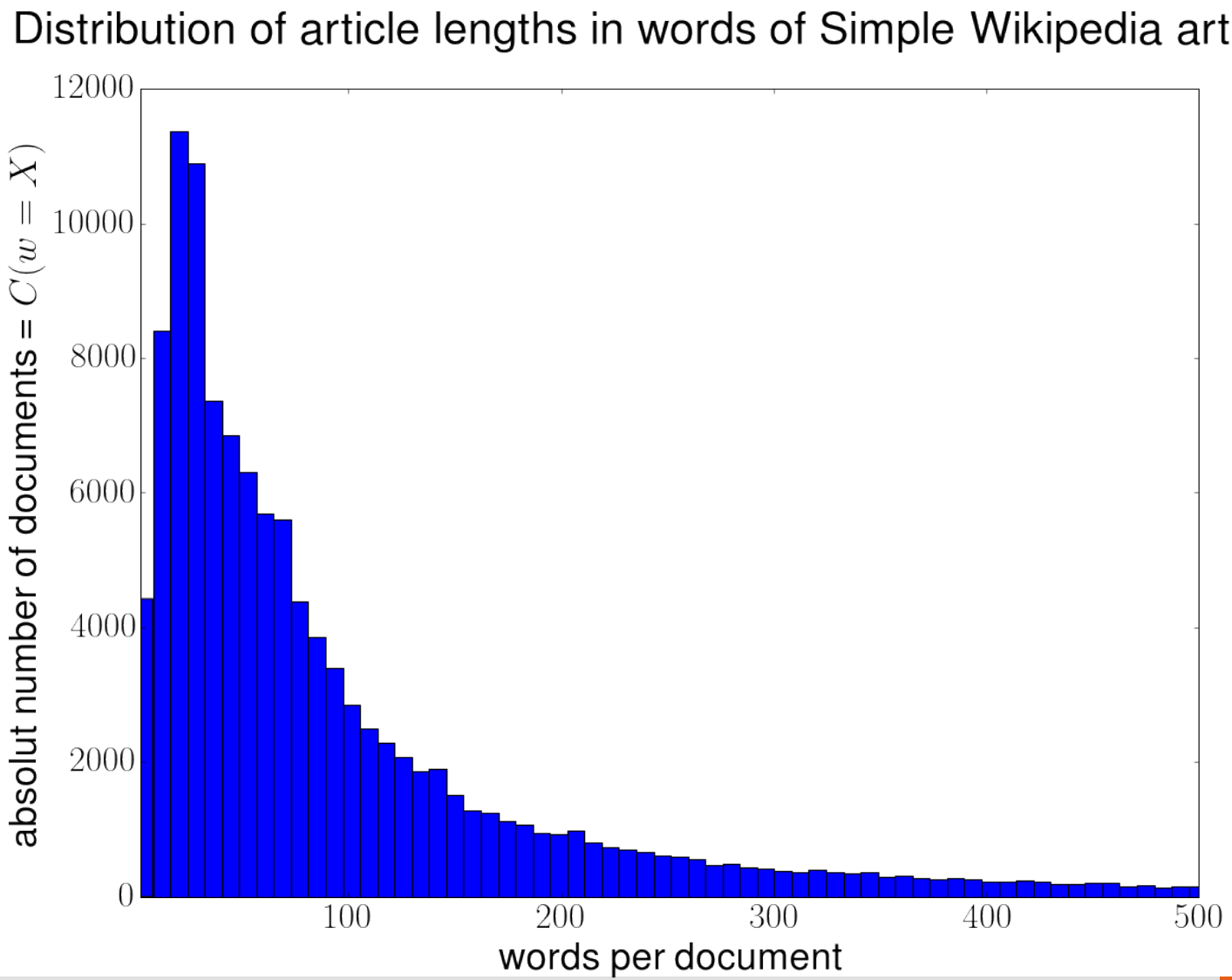
- Dividing these numbers makes
  - About 137 words per document on average

- Lets have a closer look!

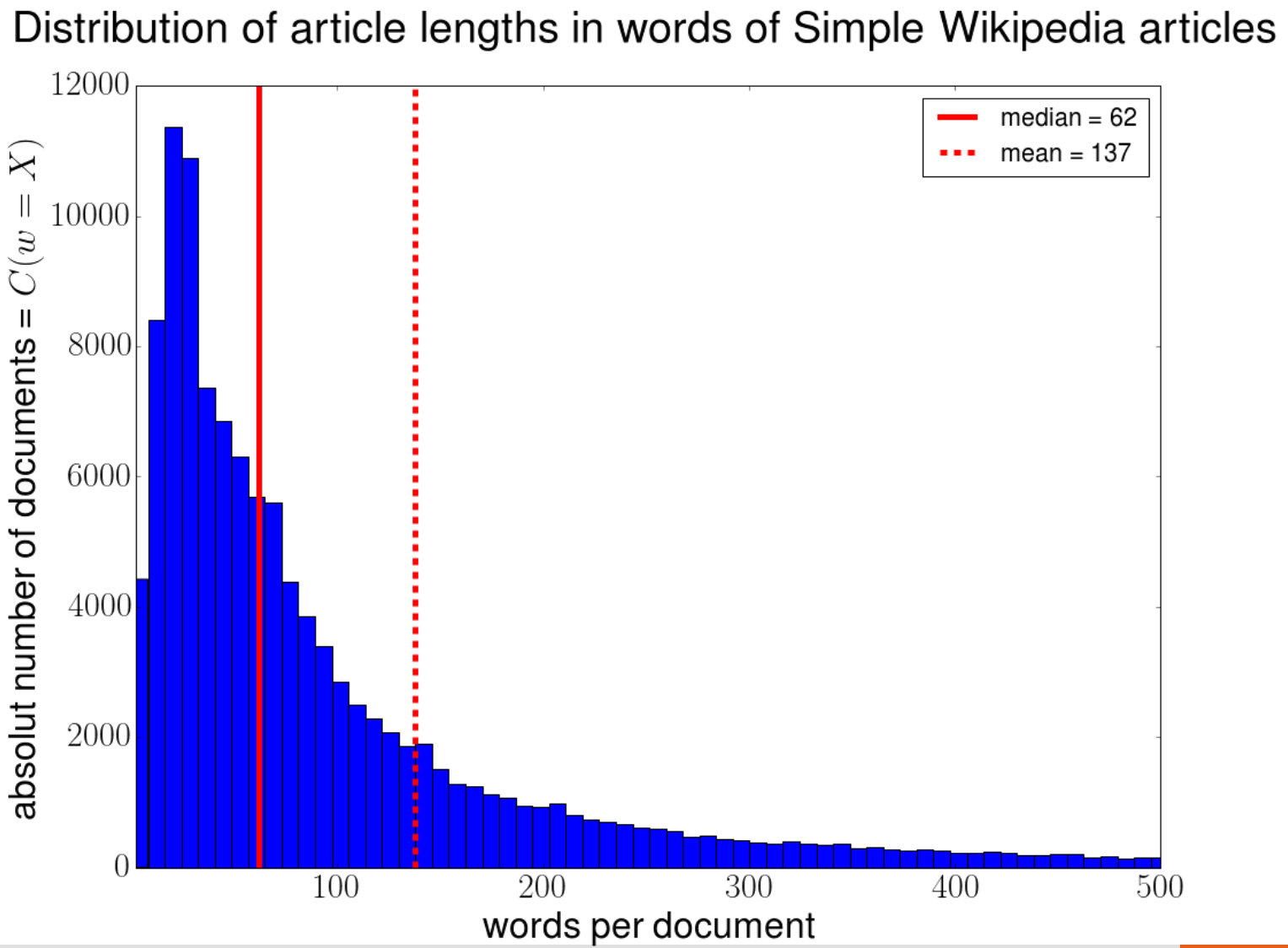
What is the typical length of a document?

- Count words for every document
- Build mean over all documents
  - 137 words per document
- Have look at the histogram
  - Visualize how many documents have
    - 0-10 words
    - 10-20 words
    - 20-30 words
    - …

Histogram of words per document



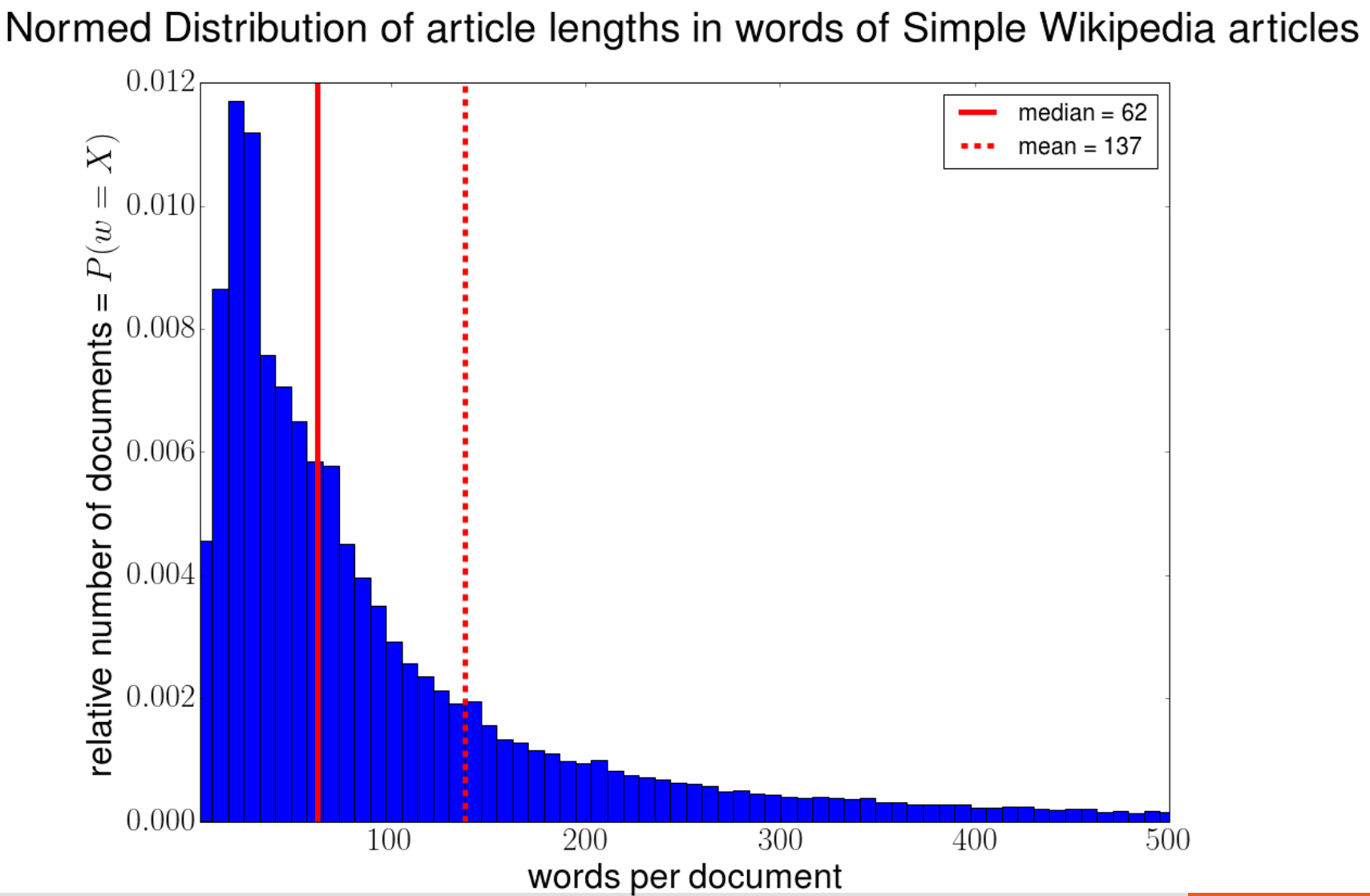
Histogram of words per document



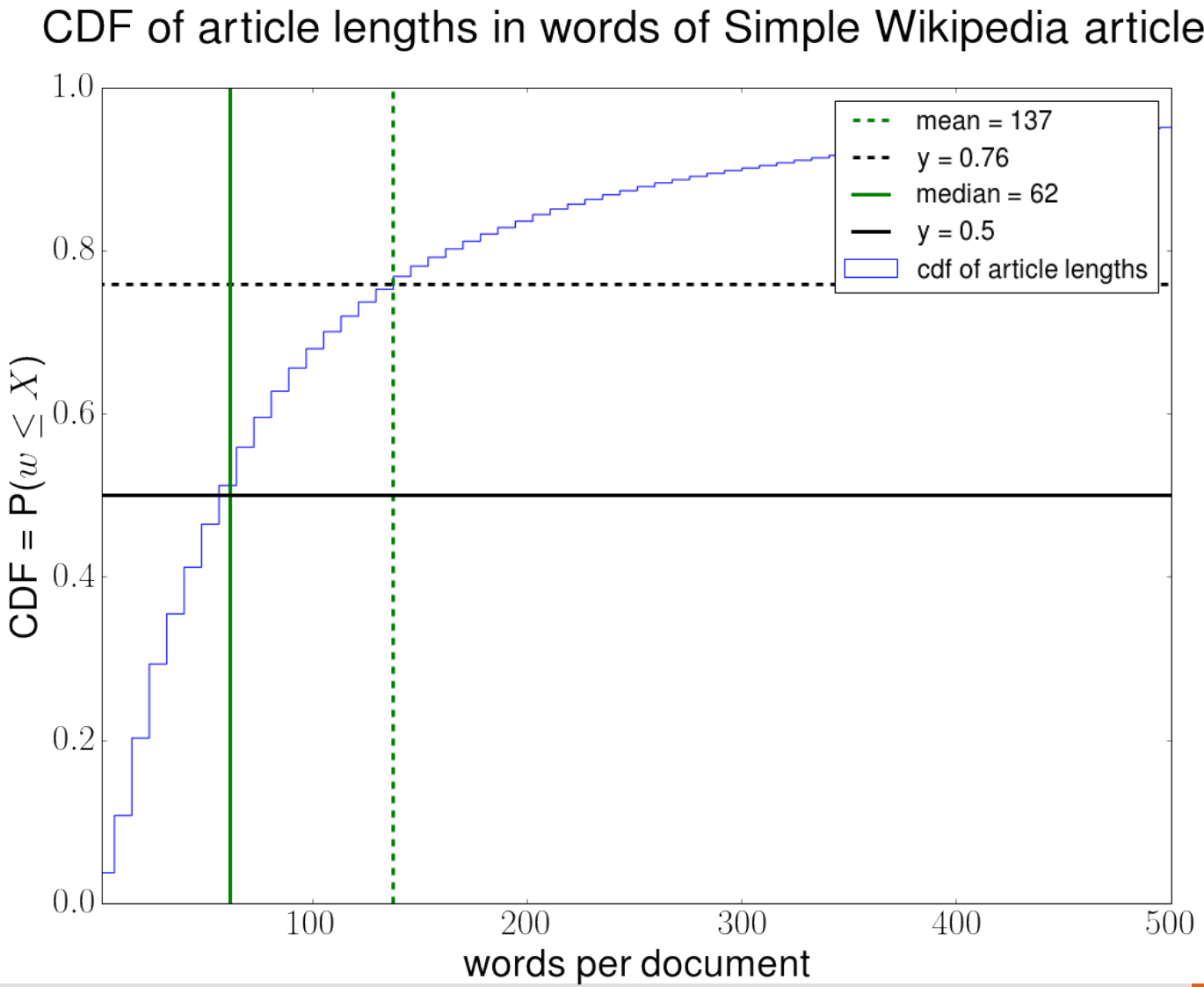
Some facts about the median

- The element which splits a set into halves of equal size
- wordsPerDoc = [10,11,12, 14, 1000000]
- mean(words) = 200’009.4
- median(words) = 12
  - Two documents have less words and two have more
- What if length(wordsPerDoc) is even?

Normalize the histogram



3 out of 4 articles are shorter than average!



Thank you for your attention!

Contact:

Rene Pickhardt
Institute for Web Science and Technologies
Universität Koblenz-Landau
rpickhardt@uni-koblenz.de

WeST
People and Knowledge Networks

Copyright:

Lesson2:
Modelling the Web with Simple Statistical Descriptive Text Models
Unit3:
Formulating a research hypothesis and finding evidence for it

Rene Pickhardt

Introduction to Web Science Part 2

Emerging Web Properties

WeST
People and Knowledge Networks

Institute for Web Science and Technologies · University of Koblenz-Landau, Germany

Completing this unit you should

- Understand the ongoing, cyclic process of research
- Know what falsifiable means and why every research hypothesis needs to be falsifiable
- Be able to formulate your own research hypothesis

First: Start with an observation

- There is English Wikipedia
- There is Simple English
- The purpose of Simple English Wikipedia is to be easier to understand and therefore more accessible than English Wikipedia

Second: Be critical and curious

- The purpose of Simple English Wikipedia is to be easier to understand and therefor more accessible than English Wikipedia

- Ask yourself: Is this really true?
  - Of course, the purpose is true

- But what about the goal?
  - Is it achieved?
  - Is it really easier to understand?

Third: Transform your question and observations into an hypothesis

- Research - Hypothesis:

  Simple English Wikipedia is easier to understand than English Wikipedia!

Some thoughts on scientific methodology

- Recall our Research – Hypothesis:
  Simple English Wikipedia is easier to understand than English Wikipedia!

- This hypothesis is **falsifiable**

- Once we find a hint why this hypothesis is not true it is falsified

- Every sound research hypothesis has this property of being falsifiable

- C.f. Karl Popper

Fourth: Develop Testable Predictions

- This is most probably the point where modeling comes into play

Testable Predictions:

- Less words are needed to understand a larger fraction of Simple English Wikipedia than English Wikipedia
  - This is a simple counting exercise

- Overall the sentences in Simple English Wikipedia are shorter and use shorter words than the ones in English Wikipedia
  - Another simple counting exercise

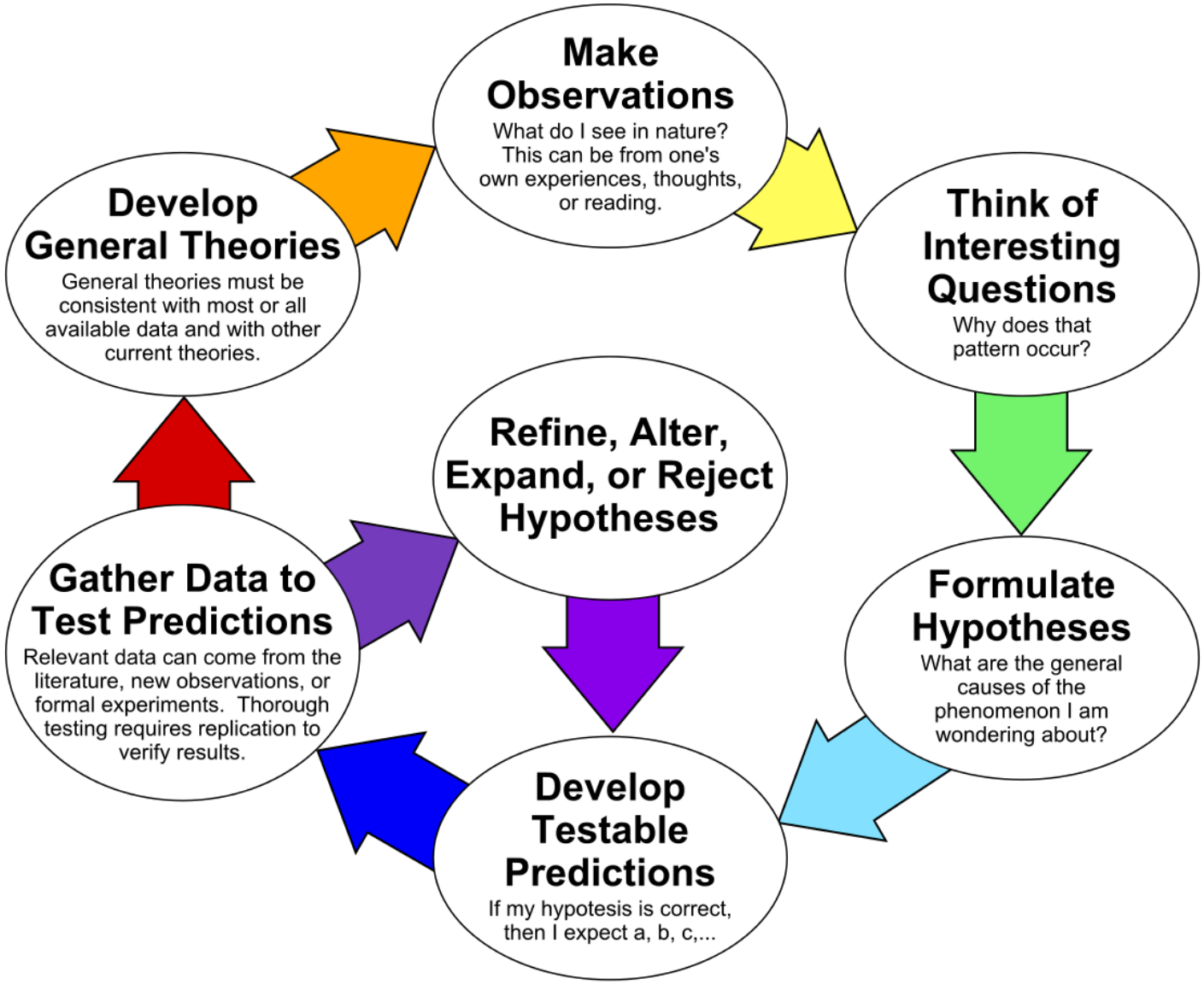
Fifth: Gather data to test predictions

- Often very difficult for the following reasons:
- Data might be in “silos” if private companies own it
  - Interesting research questions could be answered on Facebook data but it is not accessible
- Data needs to be created by asking people
  - To participate in a user study
  - Fill out questionnaires
- One of the reasons we work with Wikipedia
  - The data is available and open
  - It is just an awesome playground for research
  - It is limited since it is not used by everybody

Now we probably have to make some choice

- Either
  - Refine alter expend or reject the hypothesis
  - Go back to step 3 / 4
- Or
  - Go forward in trying to develop a general theory
  - It must be consistent with other theories and all available data
  - Often you make new observations and start over at step 1

The Scientific Method as an Ongoing Process



Roadmap for the next two units

- Analyze each of our two testable predictions
- Check if less words are needed to understand a larger fraction of Simple English Wikipedia
- See if sentences and words are really shorter
- Interpret the results and discuss them critically

Thank you for your attention!

Contact:

Rene Pickhardt
Institute for Web Science and Technologies
Universität Koblenz-Landau
rpickhardt@uni-koblenz.de

WeST
People and Knowledge Networks

Copyright:

Lesson2:
Modelling the Web with Simple Statistical Descriptive Text Models
Unit4:
Test if lesser words are required on Simple English Wikipedia to understand a larger fraction than on English Wikipedia

Rene Pickhardt

Introduction to Web Science Part 2

Emerging Web Properties

WeST
People and Knowledge Networks

Institute for Web Science and Technologies · University of Koblenz-Landau, Germany

Completing this unit you should

- Understand what a log-log plot is
- Improve your skills in reading and interpreting diagrams
- Know about the word rank / frequency plot
- Should be able to transfer a histogram or curve into a cumulative distribution function

Strategy to fulfil our test

- count the frequency of words in both corpora
- Sort the words descending to their frequency
  - This creates a ranking
- Create a plot displaying the frequency depending on the rank
- Transform this to the cumulative plot in order to test our prediction

Counting words is really simple in python

In [39]:
```
def readWordsFromWiki(filename):
    """
        opens a file which has one sentence per line (without punction marks)
        returns a list with all words
    """
    f = open(filename)
    allWords=[]
    for line in f:
        line = line[:-1]
        words = line.split()
        allWords.extend(words)
    return allWords

allSimpleWords = readWordsFromWiki("../datasets/simpleWikiAbstractsOneScentencePerLine")
allEnWords = readWordsFromWiki("../datasets/enWikiAbstractsOneScentencePerLine")
```

In [40]:
```
from collections import Counter
c=Counter(allSimpleWords)
words,frequencies = zip(*c.most_common())
print words[0:10], frequencies[0:10]
```

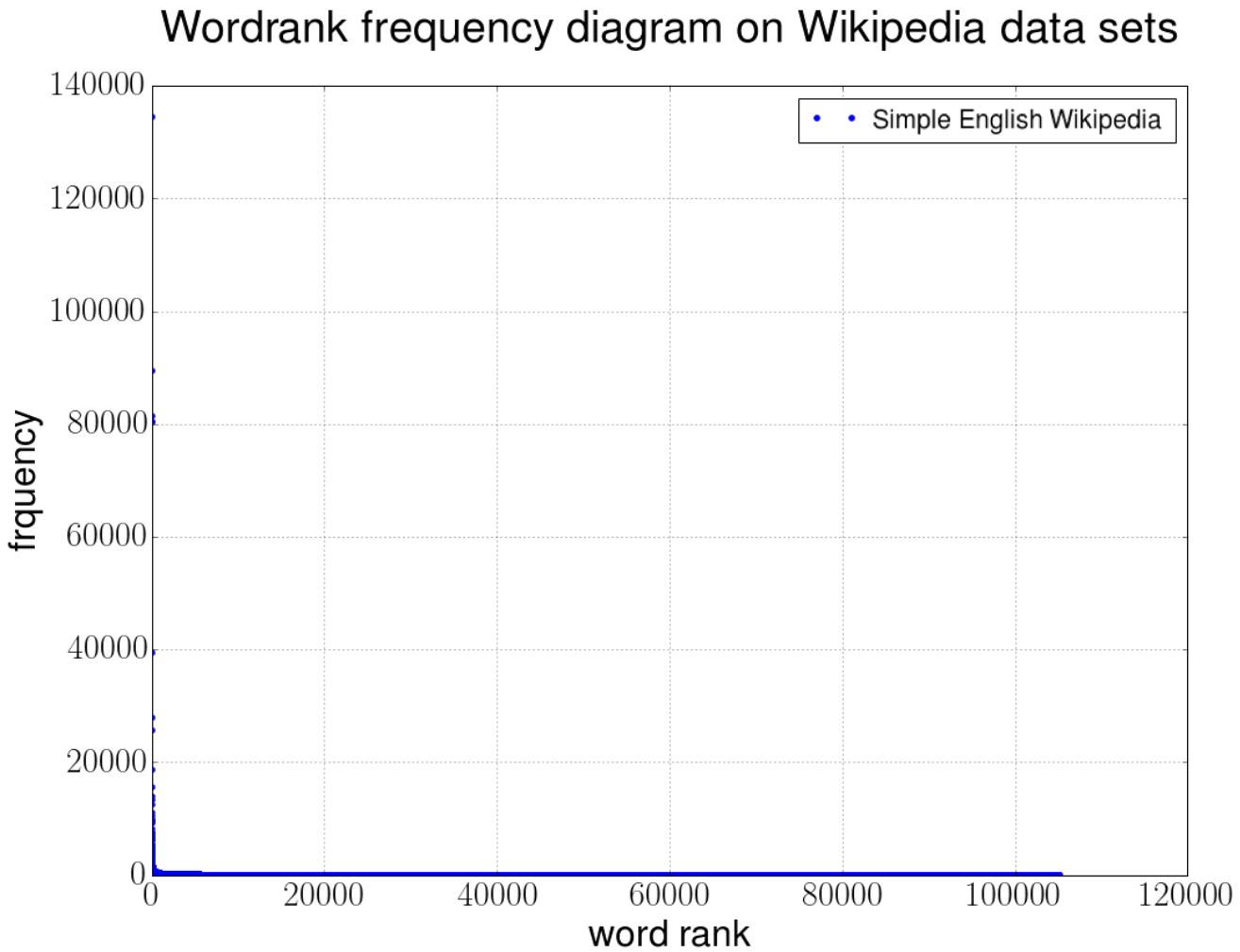
```
('the', 'is', 'a', 'of', 'in', 'and', 'it', 'was', 'to', 'an') (134415, 89447, 81349, 80376, 80309, 39475, 27820, 255
54, 18726, 15620)
```

In [23]:
```
cEn=Counter(allEnWords)
enWords,enFrequencies = zip(*cEn.most_common())
print enWords[0:10], enFrequencies[0:10]
```

```
('the', 'of', 'in', 'a', 'is', 'and', 'was', 'to', 'by', 'it') (5307042, 3247413, 2810037, 2594795, 2331626, 1983945,
 1128009, 1085090, 748863, 591726)
```

Lets look at the rank frequency diagram

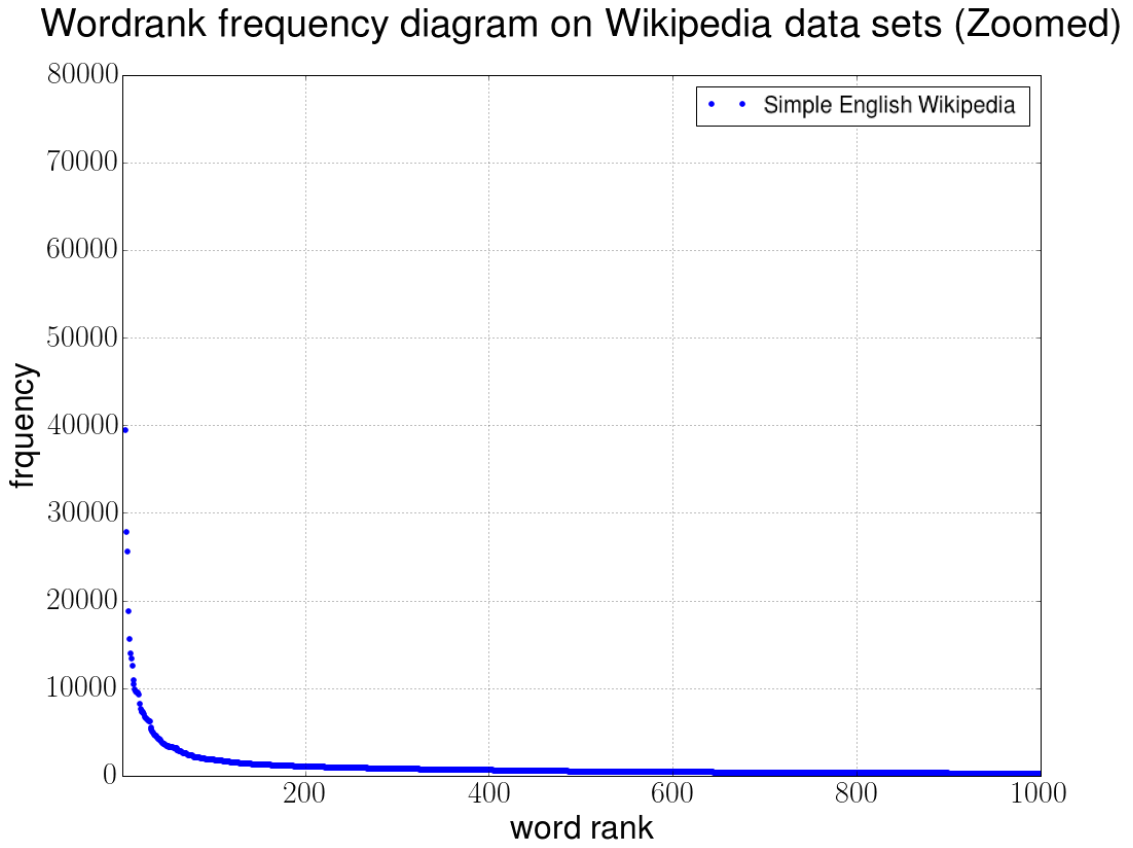
- As you can see, you see nothing (:



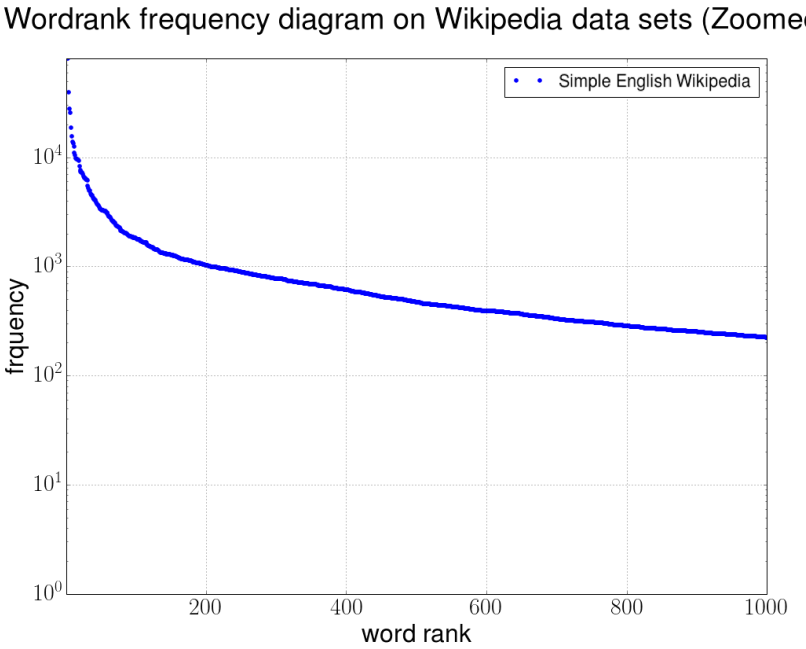
Maybe zooming the axis helps a little bit?

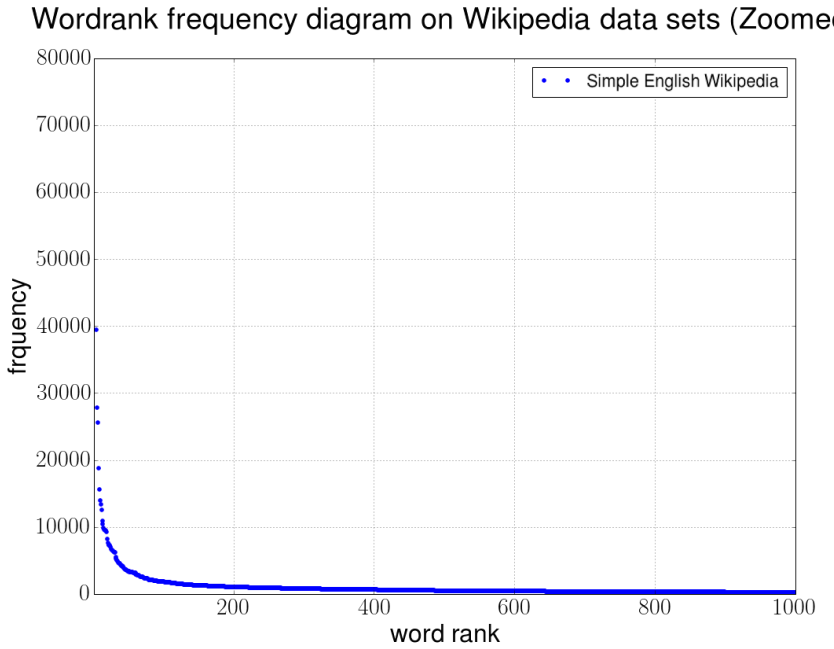
Problems with this plot:

- The frequency of words with a rank bigger than 200 can not be distinguished
- What if all ranks should be displayed?



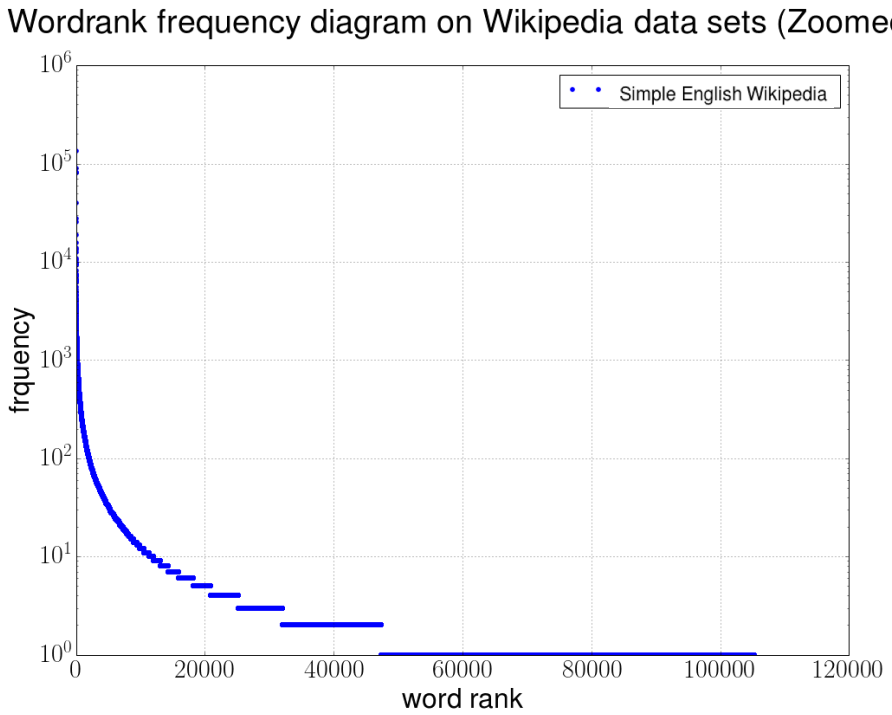
Changing the y-axis to a logarithmic scale





- Same data being used
- Very different visualization
- What happens if we include all ranks again?

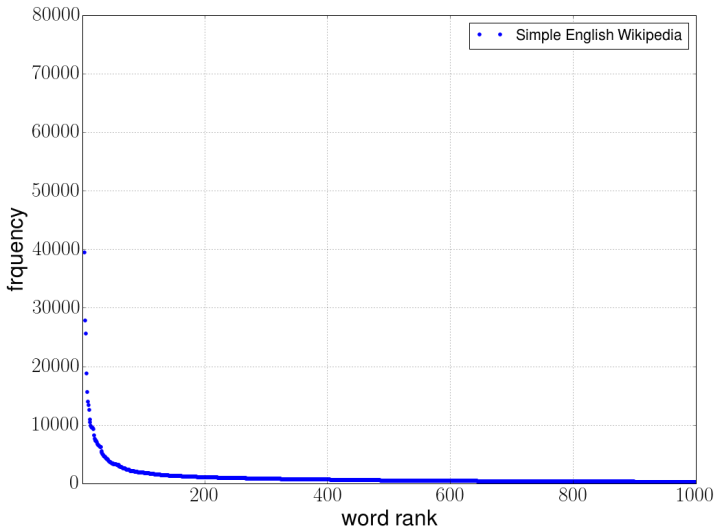
Displaying the full x-axis



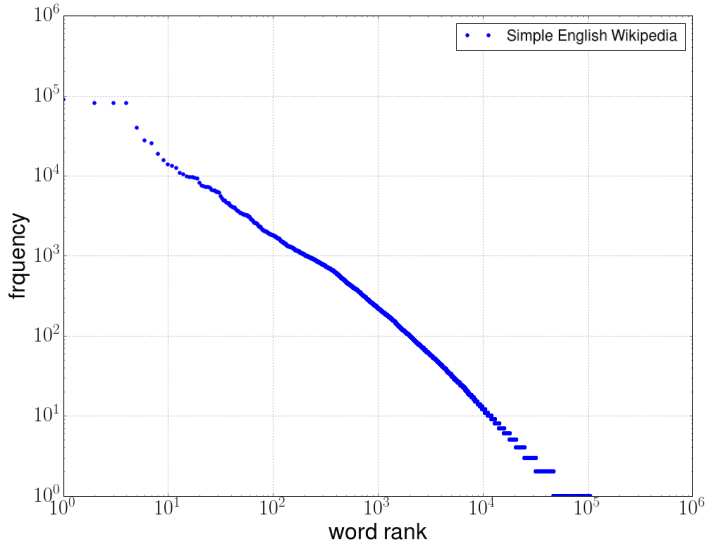
- Similar problems as before:
  - Top ranks can almost not be distinguished
- Do the same trick as before

Compare linear scale plot with log-log plot

Wordrank frequency diagram on Wikipedia data sets (Zoomed)

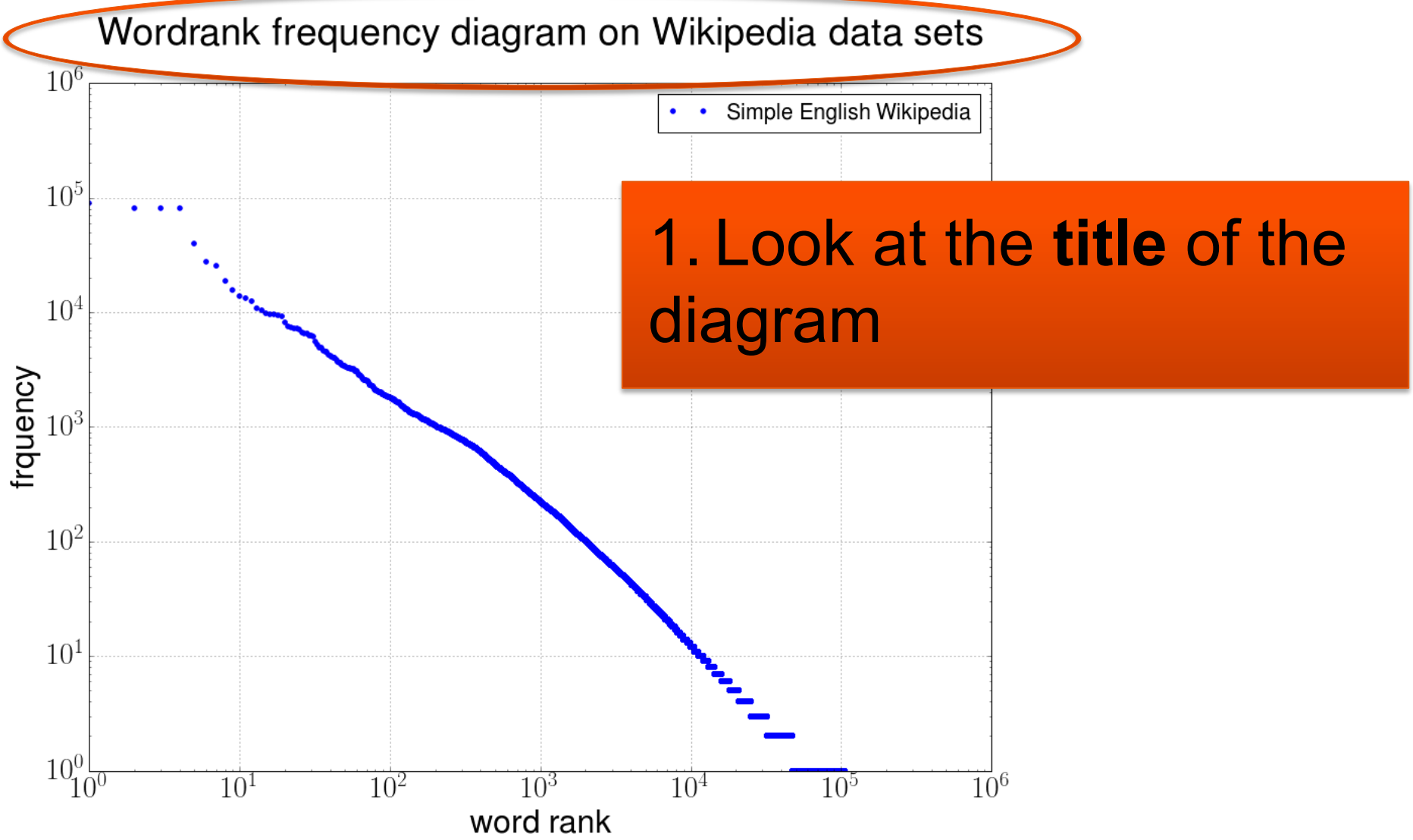


Wordrank frequency diagram on Wikipedia data sets (log-log scale)



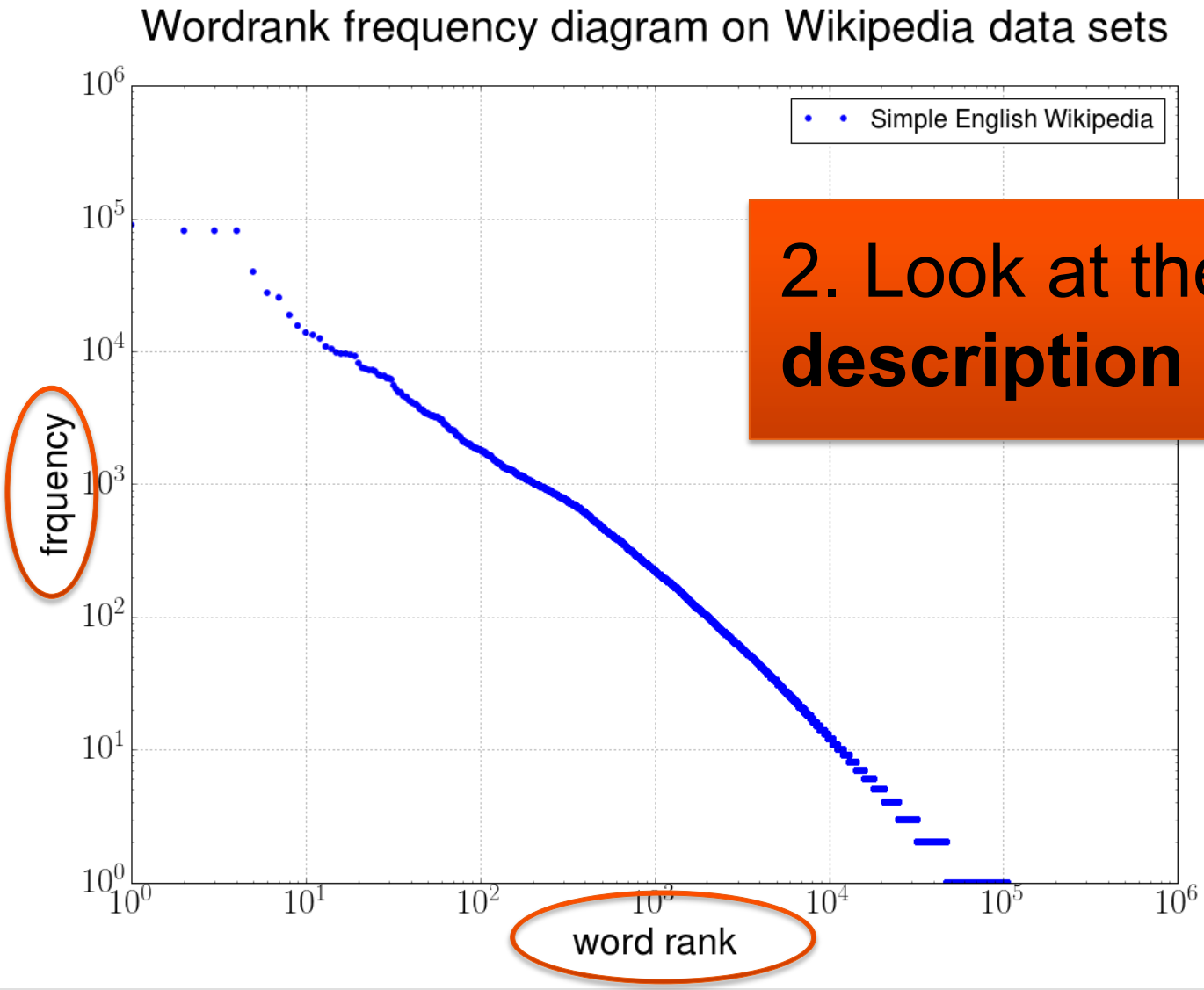
Linear	Logarithmic
Every interval displays a **fixed range** of numbers	Every interval displays one **order of magnitude**
Adding a constant number (10 k) to go from one scale unit to the next one	**Multiplying** with a constant number (in our case 10) to go from one scale to the next
Can visualize best what is happening **in a certain interval** - Usually the highest order of magnitude	Can visualize best what happens in **each** order of magnitude

6 Steps of mastering reading (log-log) plots



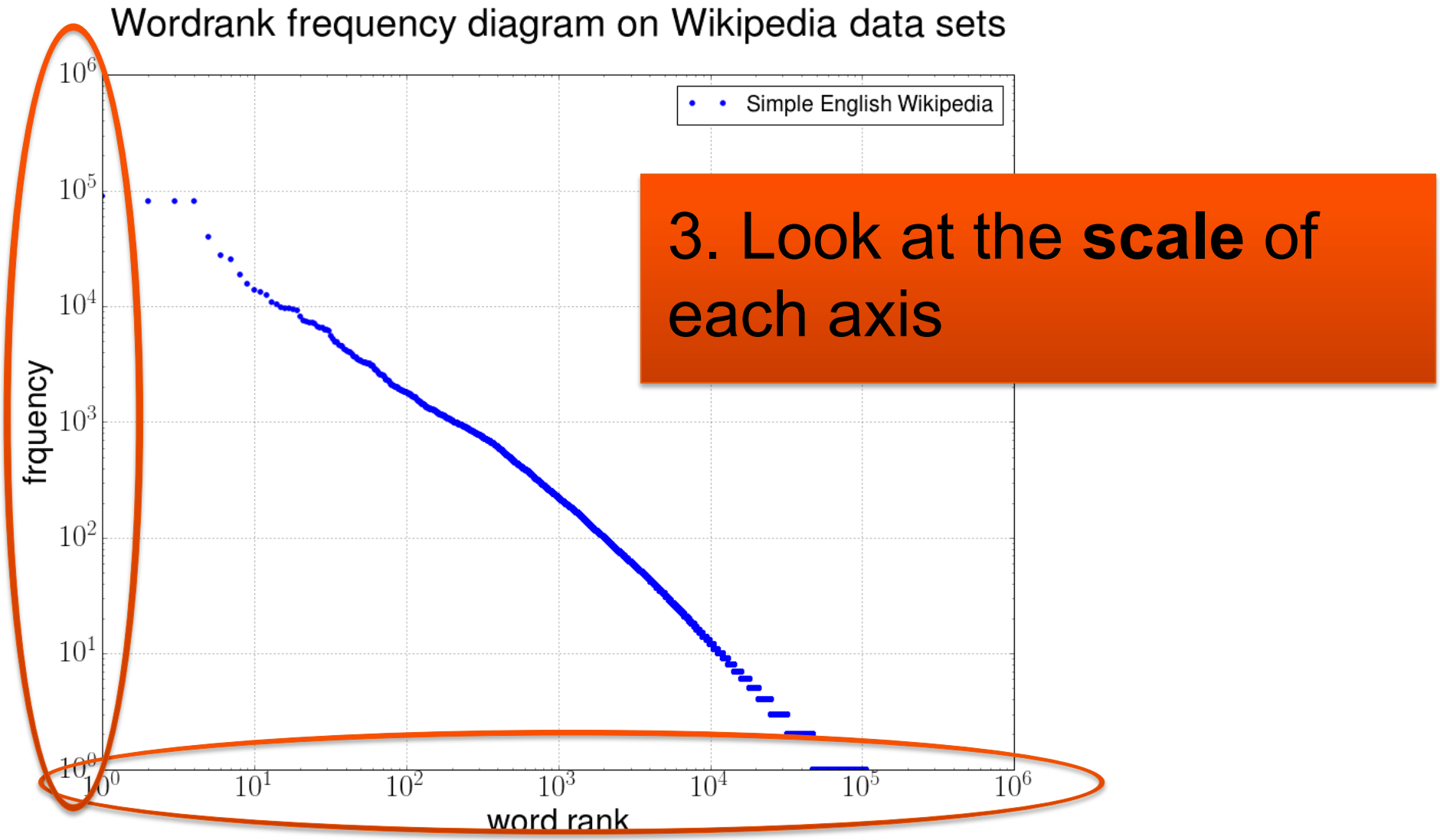
1. Look at the **title** of the diagram

6 Steps of mastering reading (log-log) plots



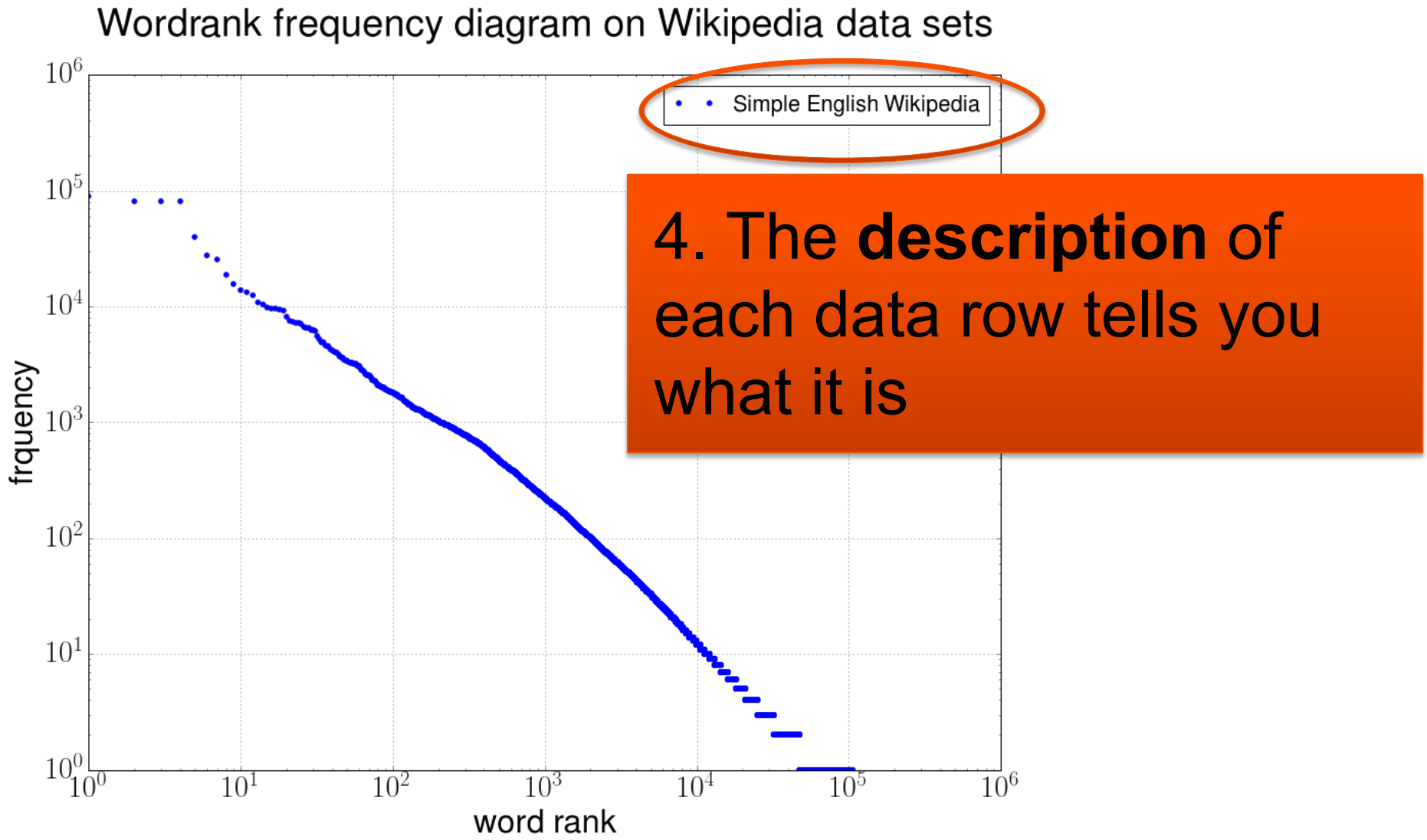
2. Look at the **description** of both axis

6 Steps of mastering reading (log-log) plots



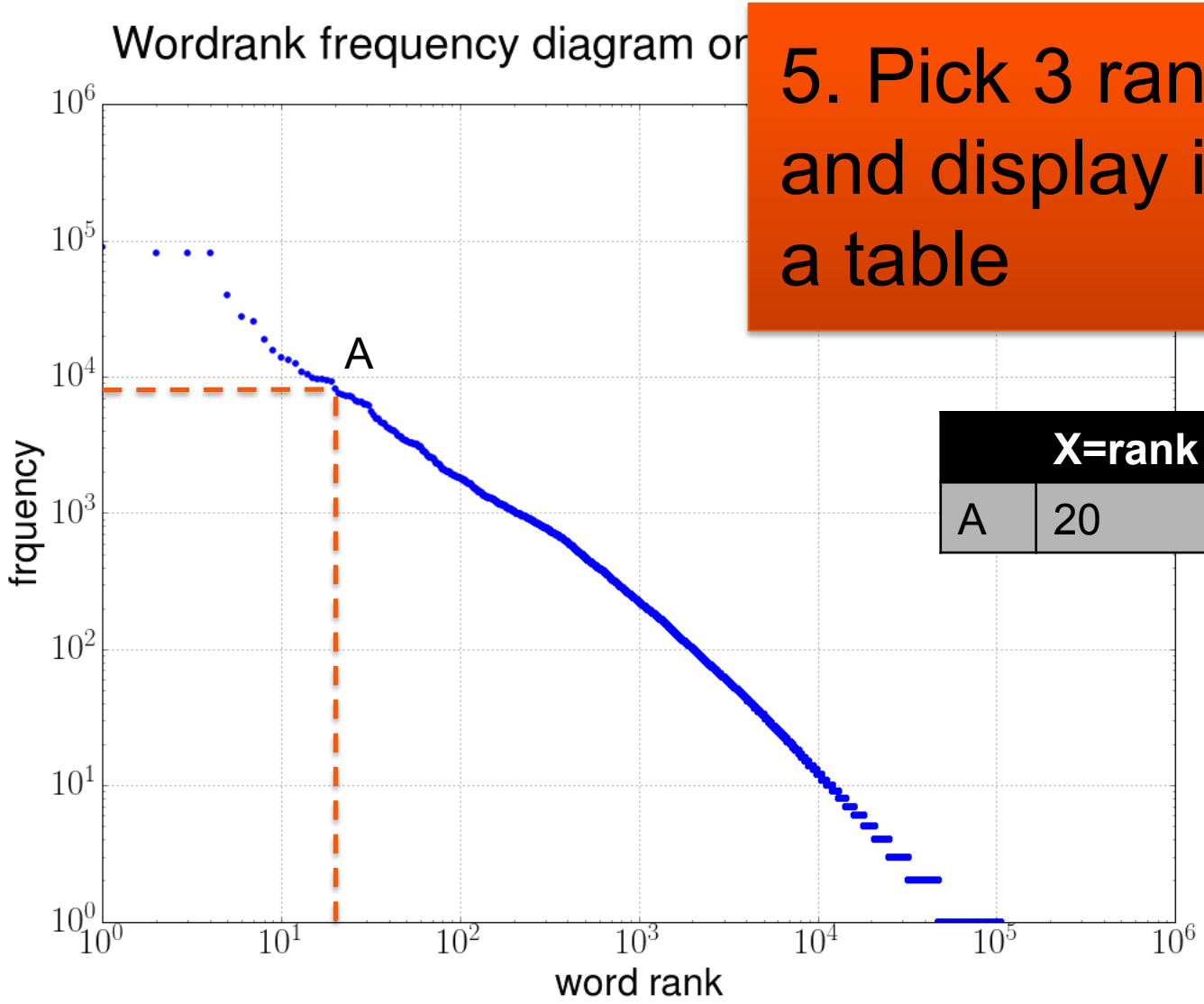
3. Look at the **scale** of each axis

6 Steps of mastering reading (log-log) plots



4. The **description** of each data row tells you what it is

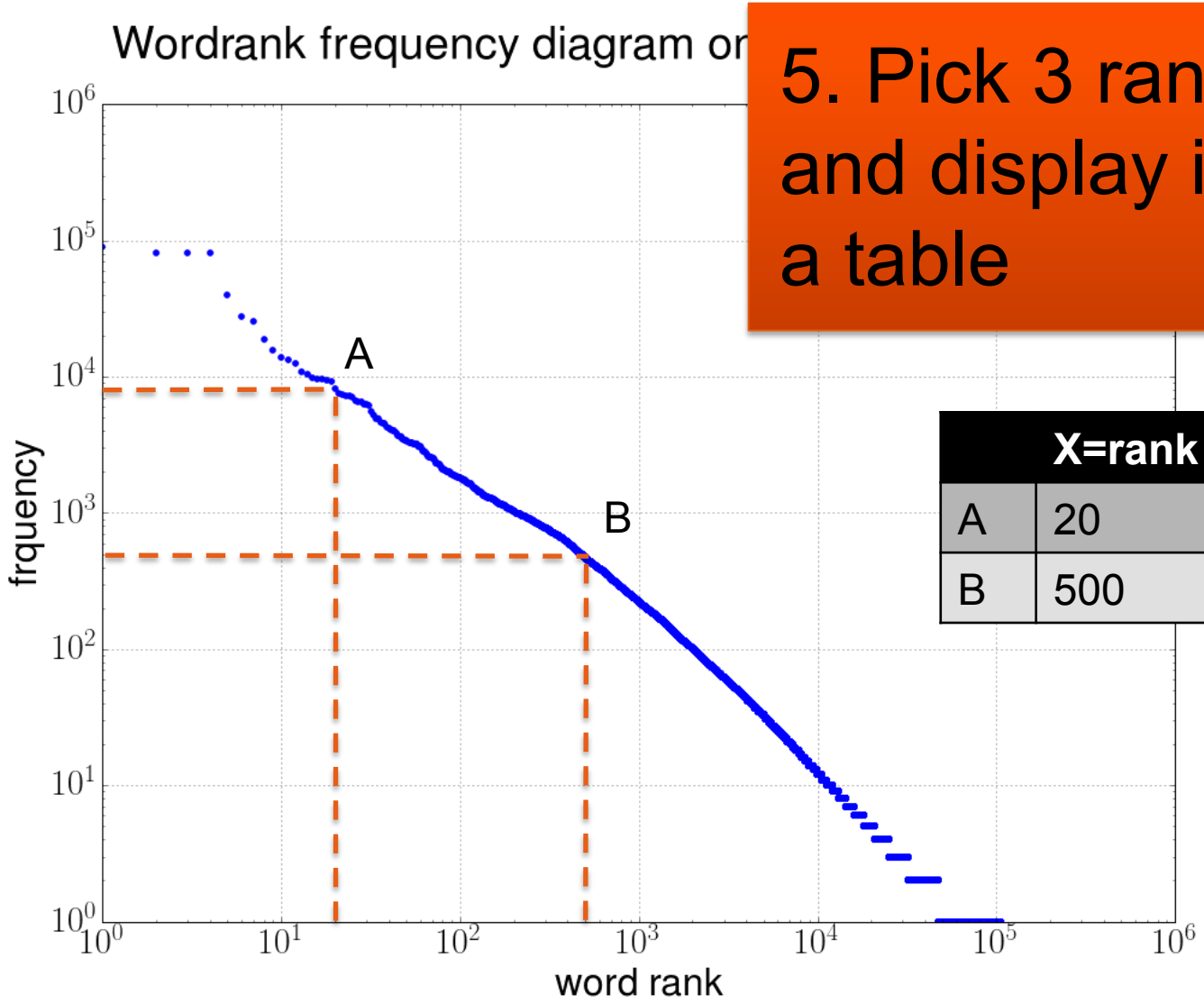
6 Steps of mastering reading (log-log) plots



5. Pick 3 random points and display its values in a table

	X=rank	Y=frequency
A	20	~8000

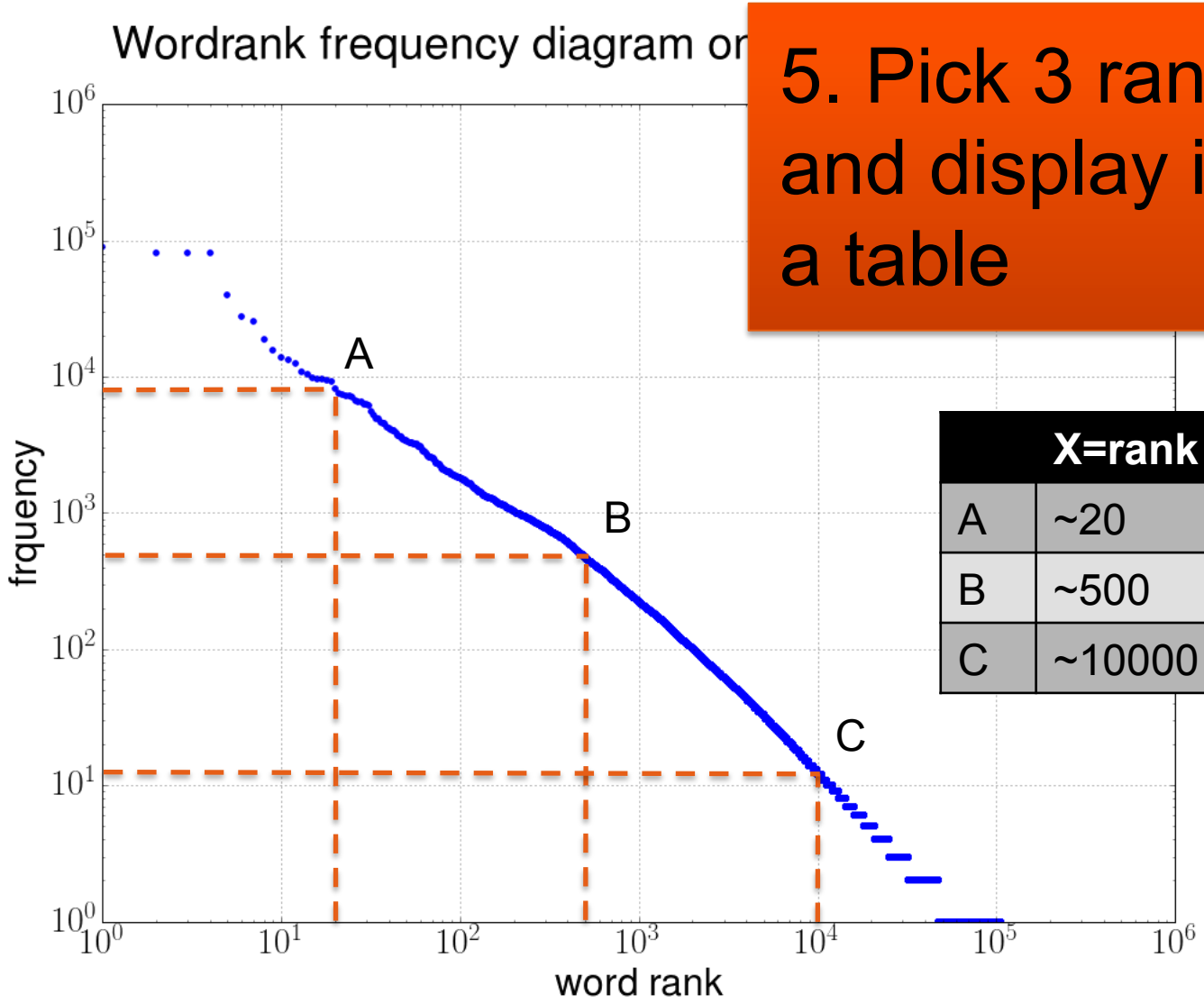
6 Steps of mastering reading (log-log) plots



5. Pick 3 random points and display its values in a table

	X=rank	Y=frequency
A	20	~8000
B	500	~500

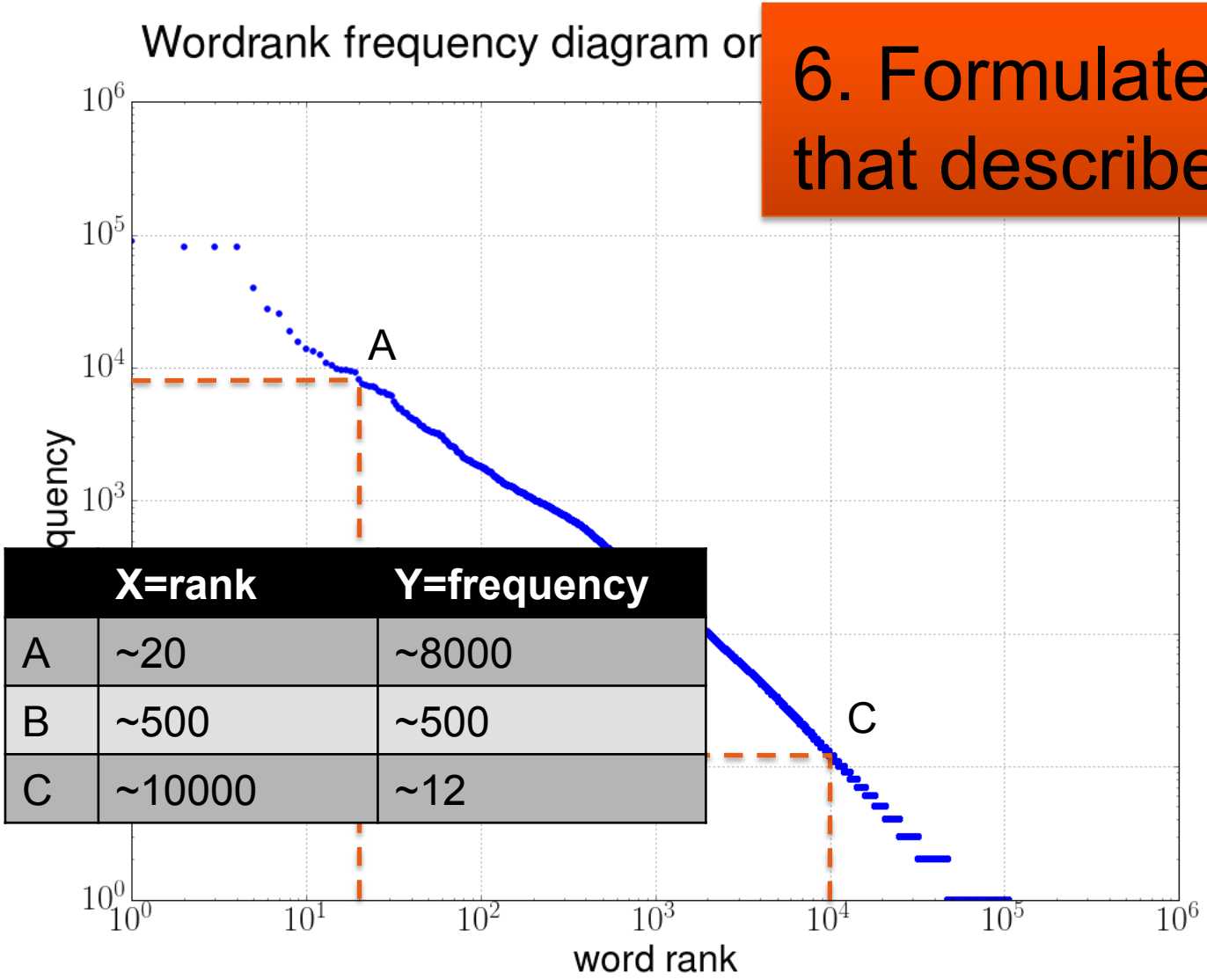
6 Steps of mastering reading (log-log) plots



5. Pick 3 random points and display its values in a table

	X=rank	Y=frequency
A	~20	~8000
B	~500	~500
C	~10000	~12

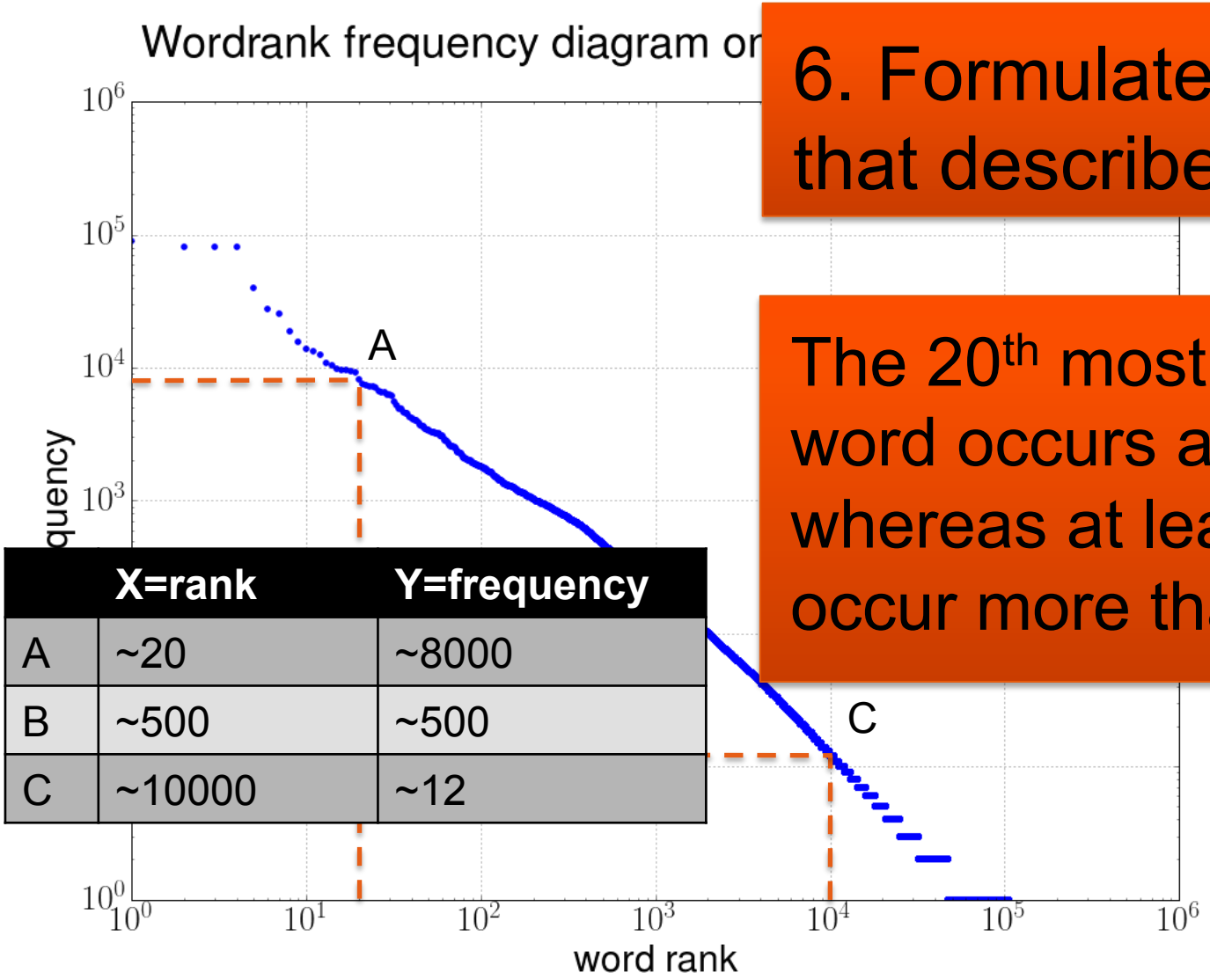
6 Steps of mastering reading (log-log) plots



	X=rank	Y=frequency
A	~20	~8000
B	~500	~500
C	~10000	~12

6. Formulate a sentence that describes the plot

6 Steps of mastering reading (log-log) plots



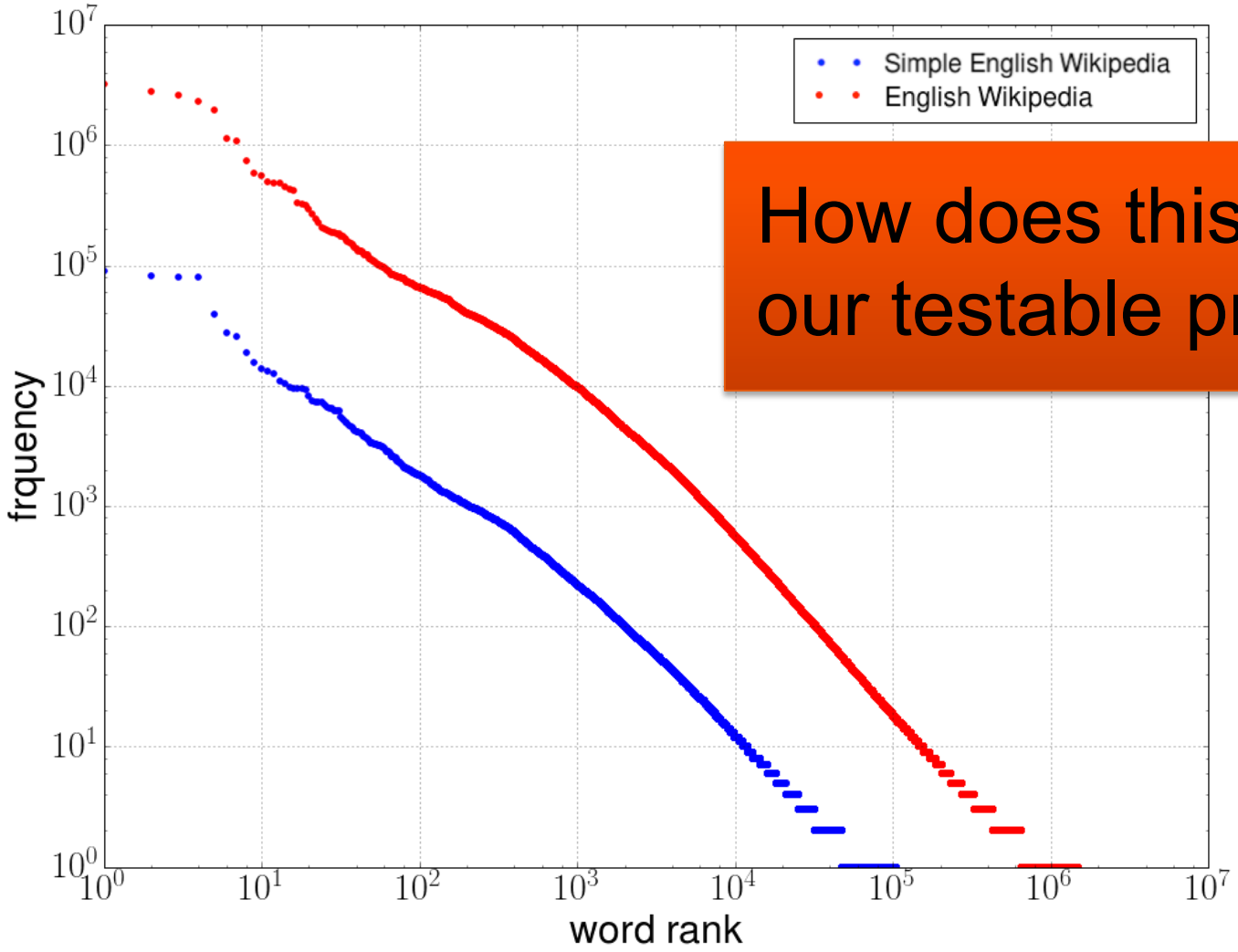
	X=rank	Y=frequency
A	~20	~8000
B	~500	~500
C	~10000	~12

6. Formulate a sentence that describes the plot

The 20th most frequent word occurs about 8k times whereas at least 10k words occur more than ten times.

Visualizing both data sets

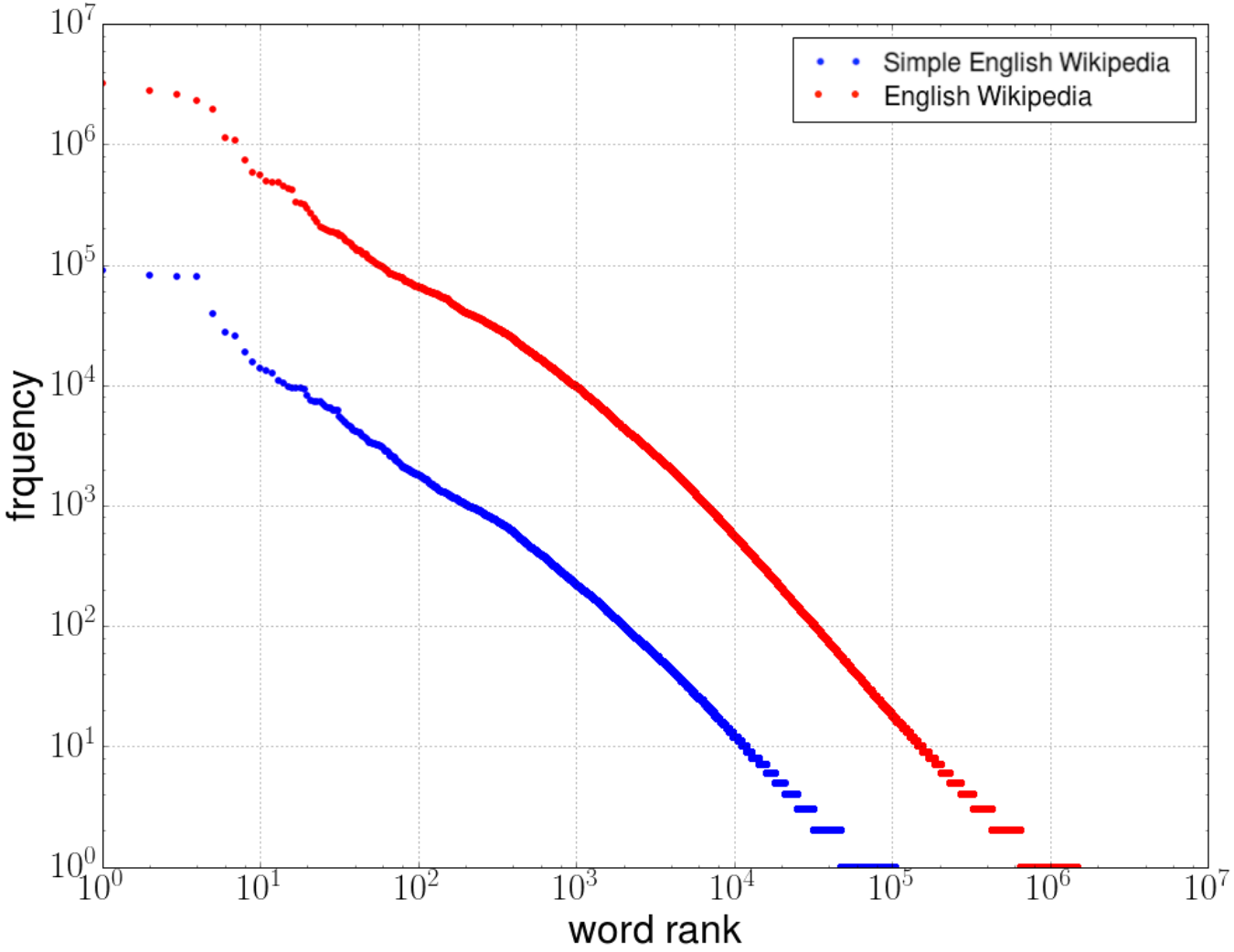
Wordrank frequency diagram on Wikipedia data sets (log-log scale)



How does this support our testable prediction?

Beware word order not the same!

Wordrank frequency diagram on Wikipedia data sets (log-log scale)



Comparing the top 10 words

	Simple English Wiki	English Wiki
1st	the	the
2nd	is	of
3rd	a	in
4th	of	a
5th	in	is
6th	and	and
7th	it	was
8th	was	to
9th	to	by
10th	an	it
Average frequency	**20.04**	**57.74**
Median frequency	**1**	**1**

Creating the Cumulative Distribution Function

In [ ]:

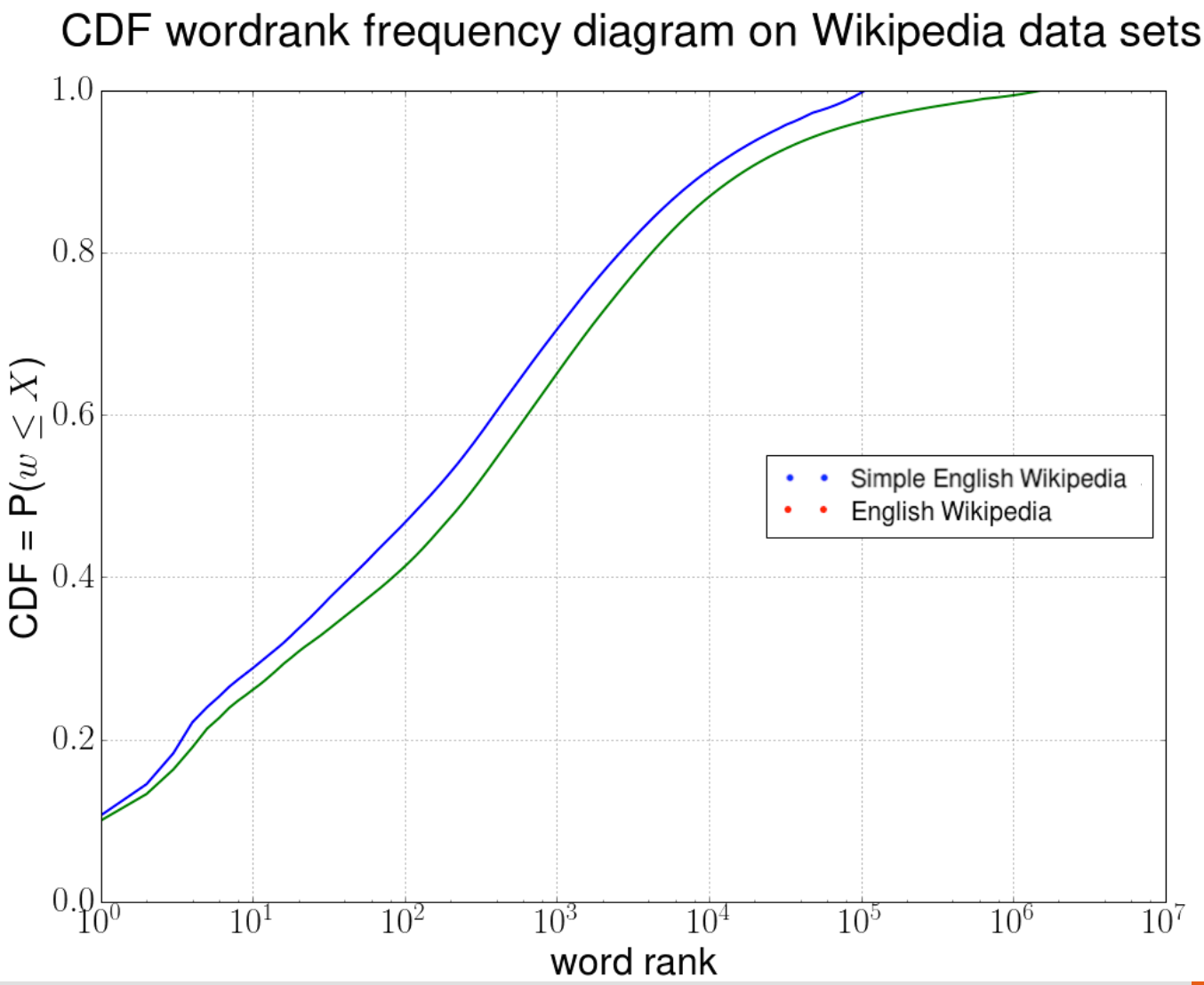
```
from collections import Counter

def getWordCDF(f):
    allWords=readWordsFromWiki(f)
    c=Counter(allWords)
    words,frequencies = zip(*c.most_common())
    cumsum = np.cumsum(frequencies)
    normedcumsum = [x/float(cumsum[-1]) for x in cumsum]
    wrank = {words[i]:i+1 for i in range(0,len(words))}
    return wrank,normedcumsum

f = open("../datasets/simpleWikiAbstractsOneScentencePerLine")
simpleWordRanks, simpleNormedCumsum = getWordCDF(f)

f = open("../datasets/enWikiAbstractsOneScentencePerLine")
enWordRanks, enNormedCumsum = getWordCDF(f)
```

Visualizing the CDF!



Now lets be critical!

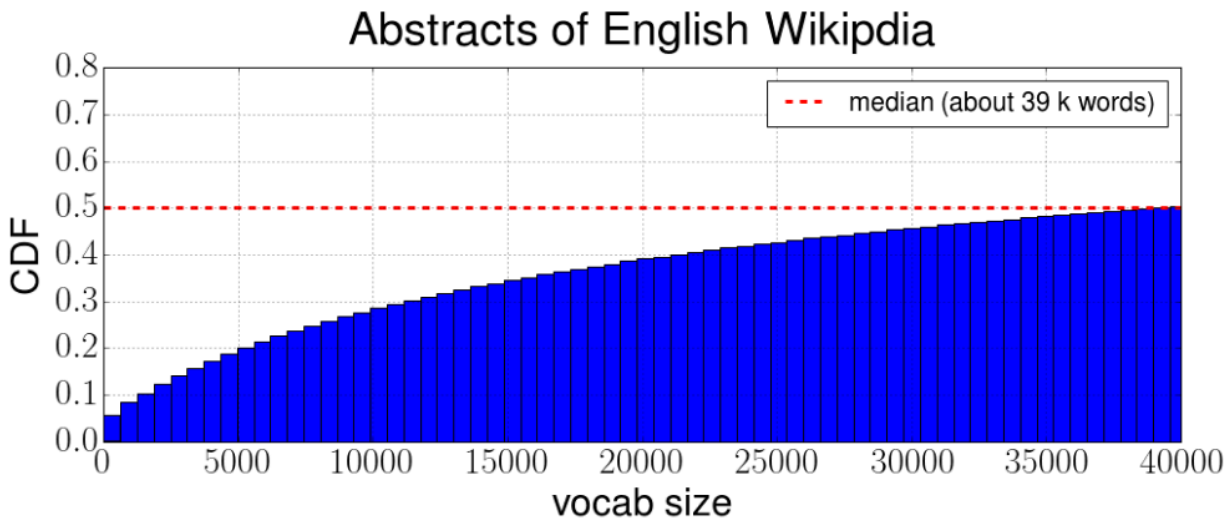
- Understanding 80% of all words does not necessarily mean that one understands 80% of the text
- Or do you understand the meaning of:
  - But it is her Schadenfreude
- English Wikipedia Corpus much bigger / more articles than Simple English
  - Comparing apples and peaches?
- Counting is ambiguous: Various forms for the “same” word like:
  - word, words
  - be, was, were, am, is
  - have, has, had

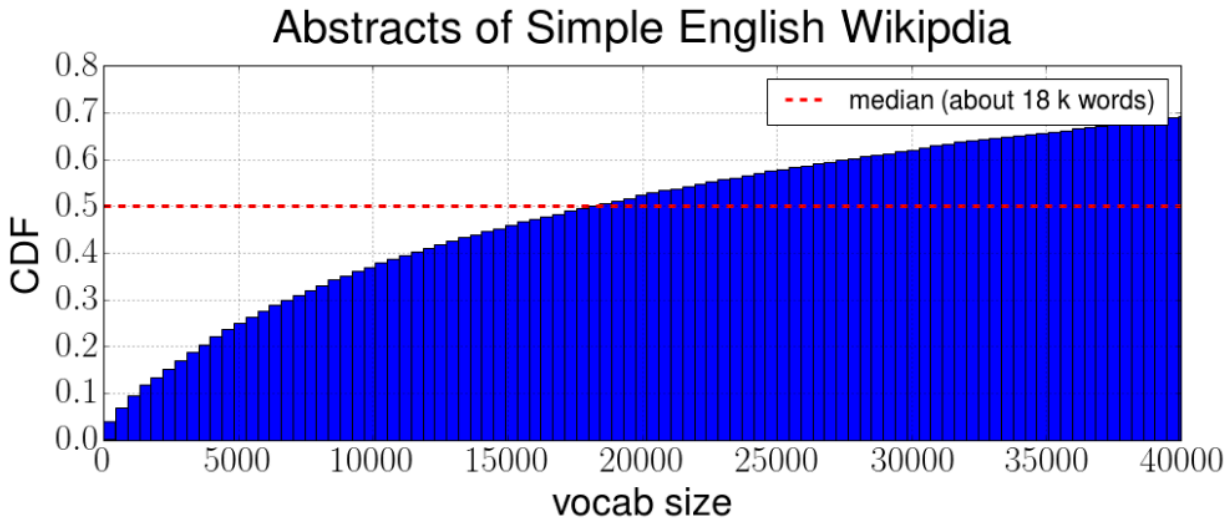
We could change the question a little bit

- How many words does one need to know all words in a given sentence?
- Can be done with the same tools and techniques
- Lets dig directly into the results

Repeat on sentences instead of words

CDF for understanding all words in a scentince given a vocab of top popular words of a certain size





Thank you for your attention!

Contact:

Rene Pickhardt
Institute for Web Science and Technologies
Universität Koblenz-Landau
rpickhardt@uni-koblenz.de

WeST
People and Knowledge Networks

Copyright:

Lesson2:
Modelling the Web with Simple Statistical Descriptive Text Models
Unit5:
Compare the sentence lengths and word lengths of Simple and English Wikipedia

Rene Pickhardt

Introduction to Web Science Part 2

Emerging Web Properties

WeST
People and Knowledge Networks

Institute for Web Science and Technologies · University of Koblenz-Landau, Germany

Completing this unit you should

- Get a feeling for interdisciplinary research
- Know the Automated Readability Index
- Have a strong sense of support for our research hypothesis
- Be able to critically discuss the limits of our models

How would linguists tackle this problem?

- Flesch-Kincaid readability test

$$fkt = 206.835 - 1.015\left(\frac{\texttt{total words}}{\texttt{total sentences}}\right) - 84.6\left(\frac{\texttt{total syllables}}{\texttt{total words}}\right)$$

- Wherever the weights and coefficients drop from the idea is clear:
  - first term is low if sentences are shorter
  - second term is low if words have fewer syllables

- Knowing syllables is a non trivial problem for a computer

- Hard to automatically calculate

Interpreting the results of the FKRT

$$fkt = 206.835 - 1.015\left(\frac{\texttt{total words}}{\texttt{total sentences}}\right) - 84.6\left(\frac{\texttt{total syllables}}{\texttt{total words}}\right)$$

Score	School Level	Notes
90 - 100	5th grade	Very easy to read for average 11 year old
80-90	6th grade	Easy to read. Conversational English for consumers
70-80	7th grade	Fairly easy to read
60-70	8th & 9th grade	Plain English. Easily understood by 13 – 15 year old students
50-60	10th to 12th grade	Fairly difficult to read
30-50	college	Difficult to read
0-30	College graduate	Very difficult to read.

Automated Readability Index

$$ari = 4.71 \left(\frac{\texttt{total characters}}{\texttt{total words}} \right) + 0.5 \left(\frac{\texttt{total words}}{\texttt{total sentences}} \right) - 21.43$$

- Wherever the weights and coefficients drop from the idea is clear:
  - first term is low if words have fewer characters
  - second term is low if sentences are shorter
- Counting words, sentences and characters is easy for a computer
- Formula corresponds to our testable prediction

Interpreting the results of the ARI

$$ari = 4.71\left(\frac{\texttt{total characters}}{\texttt{total words}}\right) + 0.5\left(\frac{\texttt{total words}}{\texttt{total sentences}}\right) - 21.43$$

Score	Age	Grade Level
1	5-6	Kindergarten
2	6-7	First grade
3	7-8	Second grade
4	8-9	Third grade
5	9-10	Fourth grade
6	10-11	Fifth grade
7	11-12	Sixth grade
8	12-13	Seventh grade
9	13-14	Eighth grade
10-13	15-18	High school
> 14	18-22	College

What does the ARI for Wikipedia look like?

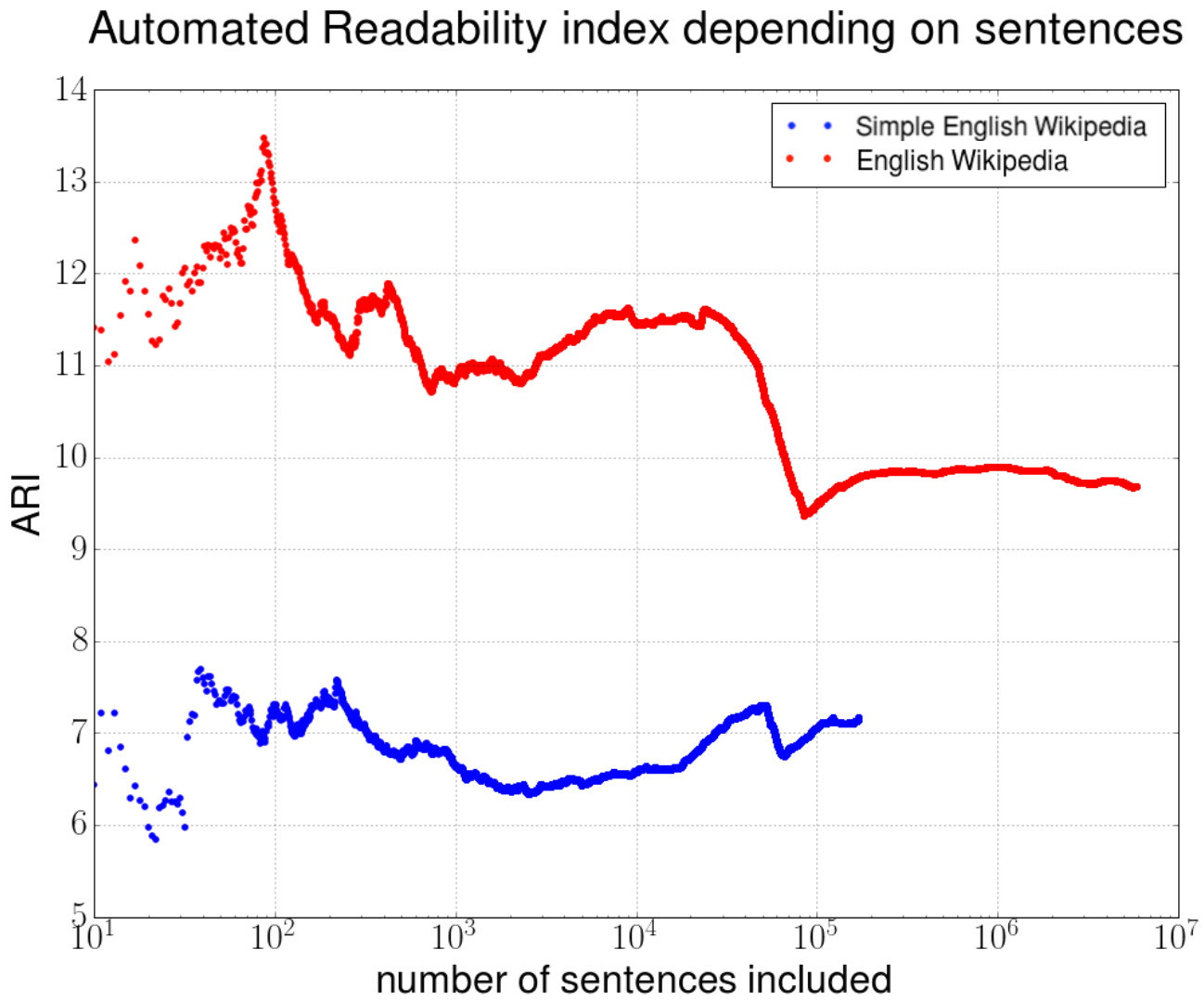
In [83]:
```
#491M    enWikiAbstractsOneScentencePerLine
# 11M    simpleWikiAbstractsOneScentencePerLine
def ari(fp):
    numSentences = 0
    numWords = 0
    numChars = 0
    for sentence in f:
        words = sentence.split(" ")
        numSentences = numSentences + 1
        numWords = numWords + len(words)
        for word in words:
            numChars = numChars + len(word)
    return 4.71*(float(numChars)/numWords) + 0.5*float(numWords)/numSentences - 21.43

f = open("../datasets/simpleWikiAbstractsOneScentencePerLine")
print "SimpleEnglish ari: " , ari(f)
f = open("../datasets/enWikiAbstractsOneScentencePerLine")
print "English Wikipedia ari", ari(f)
```

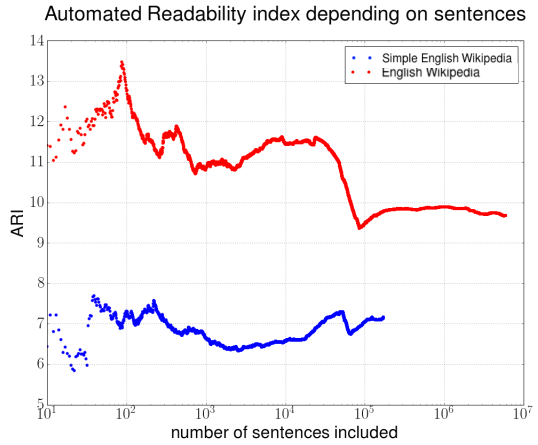
```
SimpleEnglish ari:  7.16189918182
English Wikipedia ari 9.67514555226
```

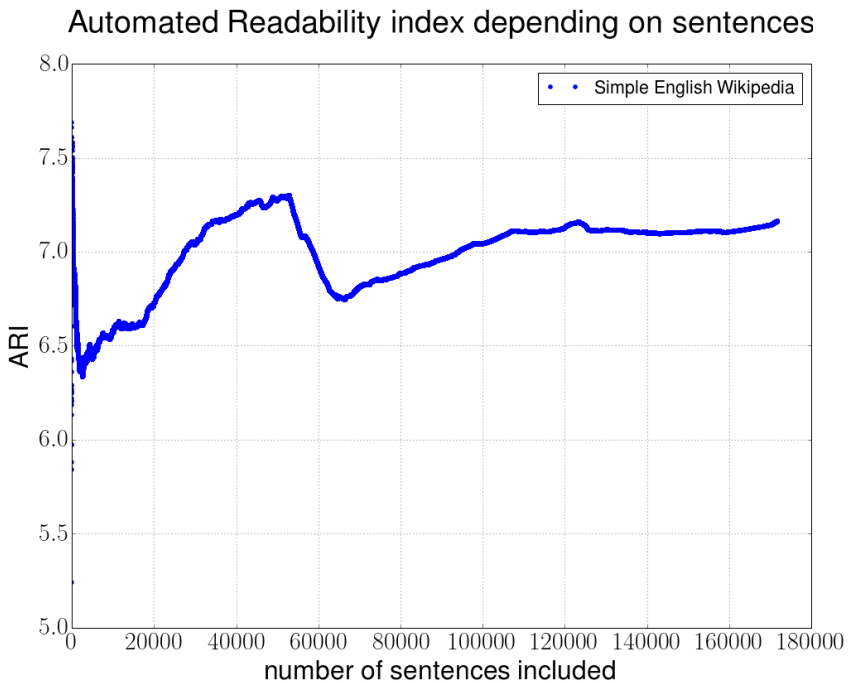
- Can we depend on the result?

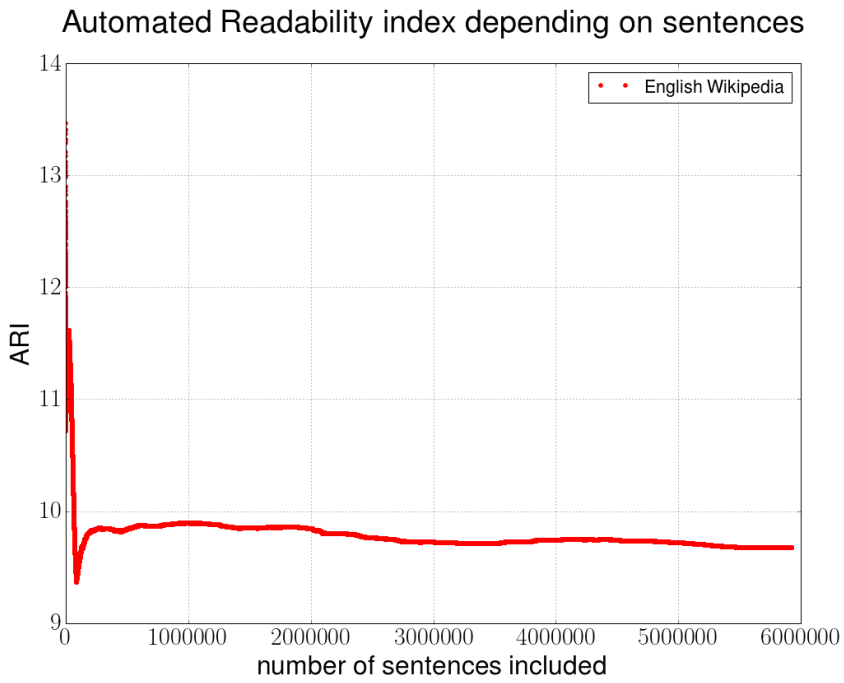
Lots of fluctuation for the readability index



Remember! Logarithmic scales are tricky







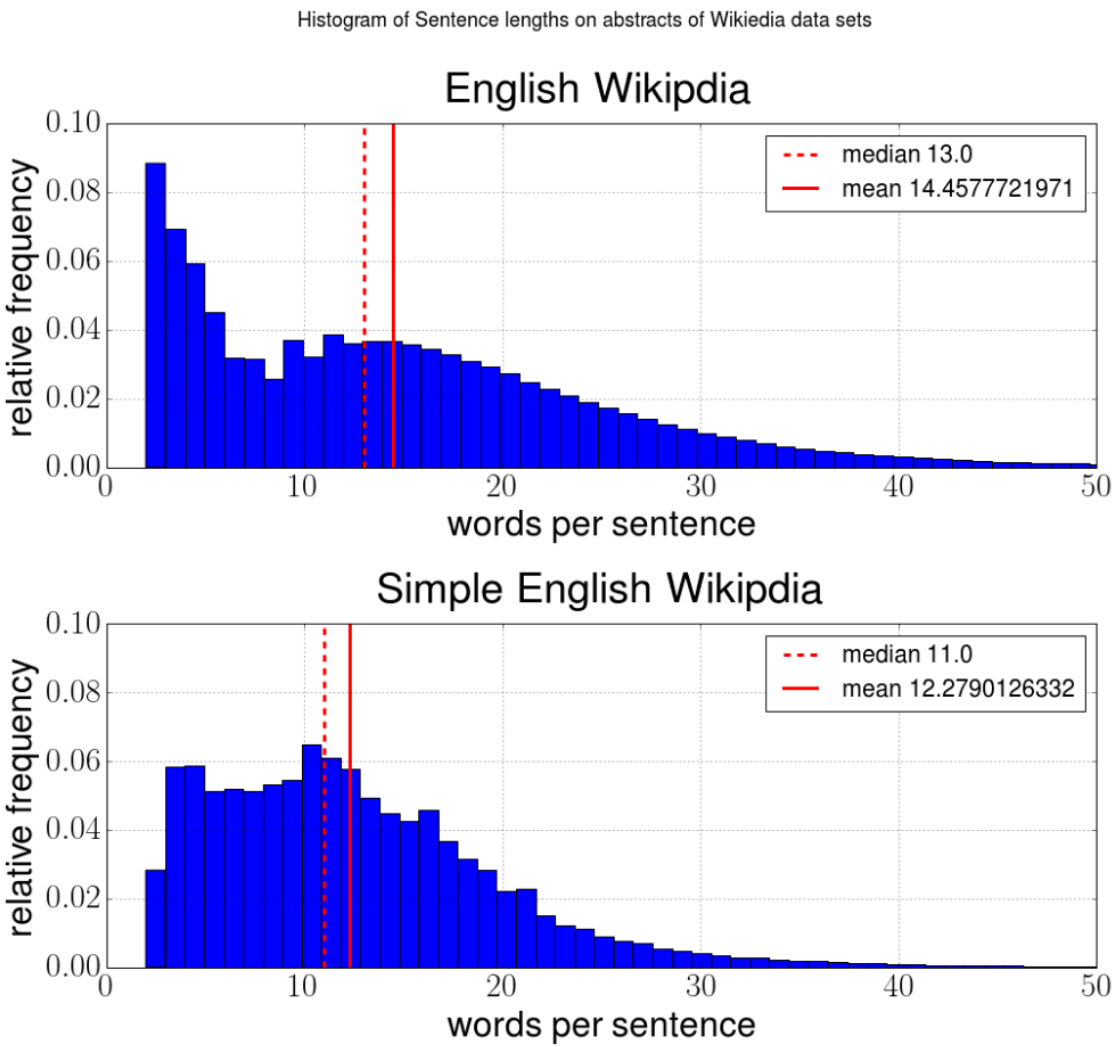
The full cycle of research… Making new observations asking questions

More words are needed to understand 50% of sentences in English Wikipedia than in Simple Wikipedia

The ARI in Simple English is lower than in English Wikipedia

Could the distribution of sentence lengths be the reason?

Research starts over again with new question



Thank you for your attention!

Contact:

Rene Pickhardt
Institute for Web Science and Technologies
Universität Koblenz-Landau
rpickhardt@uni-koblenz.de

WeST
People and Knowledge Networks

Copyright:

The instantiated model reflects a particular situation in the world

- When we take a collection of web pages in order to build a text model
- Model characterizes how the world might work in general
- But the models we study only have a special snapshot of a special situation
- also das Modell charakterisiert wie ein Ausschnitt der Welt im Allgemeinen funktioniert und das spezielle Modell instantiiert eine spezielle Situation in der Welt