Rによる自然言語研究環境の整備

石田 基広

徳島大学総合科学部

自然言語研究と統計解析

- コーパス言語学
  - 語の共起関係 (t 検定など)，テキストの指標 (多変量解析など)

自然言語研究と統計解析

- コーパス言語学
  - 語の共起関係 (t 検定など)，テキストの指標 (多変量解析など)
- 計量言語学
  - 言語単位の分布 (Zipf の法則など)

自然言語研究と統計解析

- コーパス言語学
  - 語の共起関係 (t 検定など)，テキストの指標 (多変量解析など)
- 計量言語学
  - 言語単位の分布 (Zipf の法則など)
- 心理言語学
  - 言語と認知 (誤差の処理)

自然言語研究と統計解析

- コーパス言語学
  - 語の共起関係 (t 検定など)，テキストの指標 (多変量解析など)
- 計量言語学
  - 言語単位の分布 (Zipf の法則など)
- 心理言語学
  - 言語と認知 (誤差の処理)

Rstem パッケージ　**Snowball** パッケージ

tm パッケージ　**lsa** パッケージ

zipfR パッケージ　**languageR** パッケージ　**corpora** パッケージ

コーパス言語学:トークンの処理

- Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do...
  Lewis Carroll: Alice in Wonderland.

コーパス言語学:トークンの処理

- Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do...
  Lewis Carroll: Alice in Wonderland.
- トークン化: “Alice” “was” “beginning” “to” “get” “very” “tired” “of” “sitting” “by” “her” “sister” “on” “the” “bank”...

コーパス言語学:トークンの処理

- Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do...
  Lewis Carroll: Alice in Wonderland.
- トークン化: “Alice” “was” “beginning” “to” “get” “very” “tired” “of” “sitting” “by” “her” “sister” “on” “the” “bank”...
- R でも不可能ではない．また **tm** パッケージ がサポート
- H. Baayen “Analyzing Linguistic Data” 2008 + **languageR** パッケージ
- We can make a frequency spectrum within **R**. (This is feasible only with texts or small corpora withe less than a million words.)

コーパス言語学:トークン分割

Alice was beginning to get very tired of sitting by her sister on the bank,..

```
# 連続テキストのベクトル化
alice.raw <- readLines ("alice.txt")
alice.vec <- strsplit(alice.raw,
    split = "[[:blank:]]|[[:punct:]]" ,
    extended = TRUE, perl = TRUE)
alice <- alice.vec[alice.vec != “”]
```

“Alice” “was” “beginning” “to” “get” “tired” “of” “sitting” “by” “her” “sister” “on” “bank”...

コーパス言語学:トークン分割

Alice was beginning to get very tired of sitting by her sister on the bank,..

```
# 連続テキストのベクトル化
alice.raw <- readLines ("alice.txt")
alice.vec <- strsplit(alice.raw,
    split = "[[:blank:]]|[[:punct:]]" ,
    extended = TRUE, perl = TRUE)
alice <- alice.vec[alice.vec != “”]
```

“Alice” “was” “beginning” “to” “get” “tired” “of” “sitting” “by” “her” “sister” “on” “bank”...

166 万語のテキストの場合 R (on ESS/Emacs) で約 6 秒, Java (on Eclipse) で約 1 秒 (環境: Core2Duo, Ubuntu7.10)

Rでテキスト処理

トークン化に続く処理

```
library(tm) # Text Mining パッケージ
# 余分な空白類を削除
alice.DC2 <- tmMap (alice.DC, stripWhitespace)
# stopWords (and, or, ...) を削除
alice.DC3 <- tmMap (alice.DC2, removeWords, stopwords (“english”))
```

Rでテキスト処理

トークン化に続く処理

```
library(tm) # Text Mining パッケージ
# 余分な空白類を削除
alice.DC2 <- tmMap (alice.DC, stripWhitespace)
# stopWords (and, or, ...) を削除
alice.DC3 <- tmMap (alice.DC2, removeWords, stopwords (“english”))
alice.DC4 <- tmMap (alice.DC3, stemDoc)# ステミング
    library (Rstem) # 同じことで語幹に縮める処理
    wordStem (alice) # 結果を表示する
 "alic" "was" "begin" "to" "get" "veri" "tire" "of"
"sit" "by" "her" "sister" "on" "the"
```

R でコーパス言語学

- 語の共起関係 (collocations)
  - 有意な配列パターン
  - kick the bucket
  - take place
  - powerful computer vs. ?? strong computer

R でコーパス言語学

- 語の共起関係 (collocations)
  - 有意な配列パターン
  - kick the bucket
  - take place
  - powerful computer vs. ?? strong computer

R で (無理やり) コロケーションを抽出

```
place <- which(alice == "place")
for(i in place){     x <- i - 2; y <- i + 2
    if (x< 0) x <- 0;      if (y > length (alice )) y <- length (alice)
    for(z in x:y){     cat ( sprintf (“%10s”, alice[z] ),”¥n” ) }}
     in          bill’s       place       for        a
     an          open         place       with       a
     at          the          place       where      it
   move          one          place       on         he
    the      dormouse’s       place       and       alice
   took          the          place       of         the
    the         whole         place       around     her
   take          the          place       of         the
```

共起関係の統計解析

- z スコア，t スコア, MI スコア
  - $z = \frac{O-E}{\sigma}$
  - $t = \frac{O-E}{\sqrt{O}}$ (by Church)
  - $MI = log_2 \frac{O}{E}$
  - **corpora** パッケージ, S.Evert

共起関係の統計解析

- z スコア，t スコア，MI スコア
  - $z = \frac{O-E}{\sigma}$
  - $t = \frac{O-E}{\sqrt{O}}$　　(by Church)
  - $MI = log_2 \frac{O}{E}$
  - **corpora** パッケージ，　S.Evert
- 正規分布，ランダムの仮定
  - t 値そのものではなく，そのランクを参照
- カイ二乗検定や対数尤度比検定
  - R ならば `loglin()` 関数など

コーパス言語学 その2

文学・文献学研究への応用

- 著者推定
  - 金明哲「自己組織化マップと助詞分布を用いた書き手の同定」2003

コーパス言語学 その2

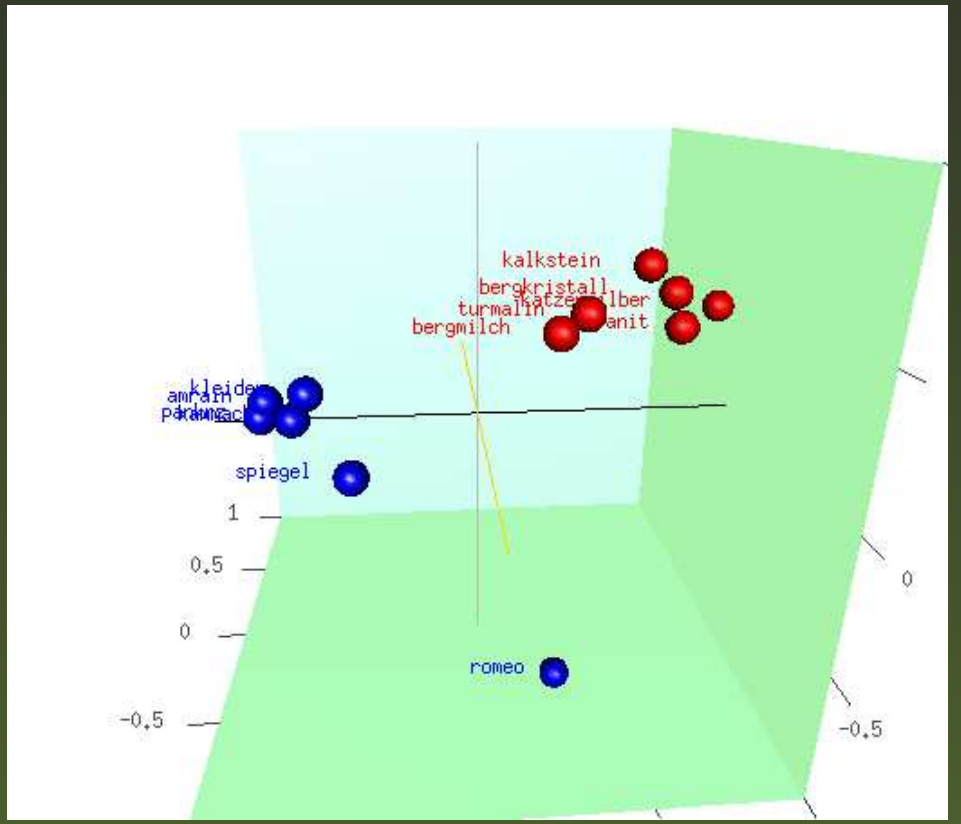
文学・文献学研究への応用

- 著者推定
  - 金明哲「自己組織化マップと助詞分布を用いた書き手の同定」2003
- 作品のターム分析
  - 田畑智司「コーパスに基づく文体論研究」2005
  - 石田基広「ベクトル空間に投射した作品の意味構造」2005

コーパス言語学 その2

文学・文献学研究への応用

- 著者推定
  - 金明哲「自己組織化マップと助詞分布を用いた書き手の同定」2003
- 作品のターム分析
  - 田畑智司「コーパスに基づく文体論研究」2005
  - 石田基広「ベクトル空間に投射した作品の意味構造」2005



lsa パッケージによるテキスト処理

- 潜在的意味インデキシング解析 (LSA)
- lsa_0.57 by Fridolin Wild
  - ディレクトリ内の全テキスト読込
  - オプションで stopwords を削除
  - stemming に対応 (**Rstem** による処理)
  - ターム・文書行列の生成 (重み付け)
  - 特異値分解
  - 文書どうしの類字度 (コサイン距離等) の算出
  - 新規検索タームの文書行列との類似度計算

ベンチマークによる試行例

- 九つのテクニカルメモのタイトルを利用
  - Scott C. Deerwester et al., Indexing by Latent Semantic Analysis, 1990
- D1 - D5 (human-computer-interaction)
- D6 - D9 (graph theory)

ベンチマークによる試行例

- 九つのテクニカルメモのタイトルを利用
  - Scott C. Deerwester et al., Indexing by Latent Semantic Analysis, 1990
- D1 - D5 (human-computer-interaction)
- D6 - D9 (graph theory)
  - D1: Human machine interface for ABC computer applications
  - D2: A survey of user opinion of computer system response time
  - D3: The EPS user interface management system
  - D4: System and human system engineering testing of EPS
  - D5: Relation of user perceived response time to error measurement
  - D6: The intersection graph of paths in trees
  - D7: Graph minors IV: Widths of trees and well-quasi-ordering
  - D8: The generation of random, binary, ordered trees
  - D9: Graph minors: A survey

ファイルの読み込みと行列作成

```
# 文書ディレクトリの指定
td <- (“/home/user/texts”)
# 必要なら stopword をロード
data(stopwords_en)
# ディレクトリを読み込み
# ターム・文書行列作成
myMatrix <- textmatrix(td,
    stopwords = stopwords_en,
    stemming = TRUE)
```

```
> myMatrix
                docs
terms            D1 D2 D3 D4 D5 D6 D7 D8 D9
  abc             1  0  0  0  0  0  0  0  0
  application     1  0  0  0  0  0  0  0  0
  comput          1  1  0  0  0  0  0  0  0
  human           1  0  0  1  0  0  0  0  0
  interfac        1  0  1  0  0  0  0  0  0
  machin          1  0  0  0  0  0  0  0  0
  opinion         0  1  0  0  0  0  0  0  0
  respons         0  1  0  0  1  0  0  0  0
  survey          0  1  0  0  0  0  0  0  1
  syst            0  1  1  2  0  0  0  0  0
```

もとの頻度行列と検索語との距離

```
myQuery <-
    query("user interface",
    rownames(myMatrix),
    stemming = TRUE)
myMat.Que <-
    cbind(myMatrix,
    myQuery)
as.matrix(round(
    cosine(myMat.Que),
    dig = 2)[,10])
```

D1	0.29
D2	0.27
D3	0.63
D4	0.00
D5	0.27
D6	0.00
D7	0.00
D8	0.00
D9	0.00

特異値分解

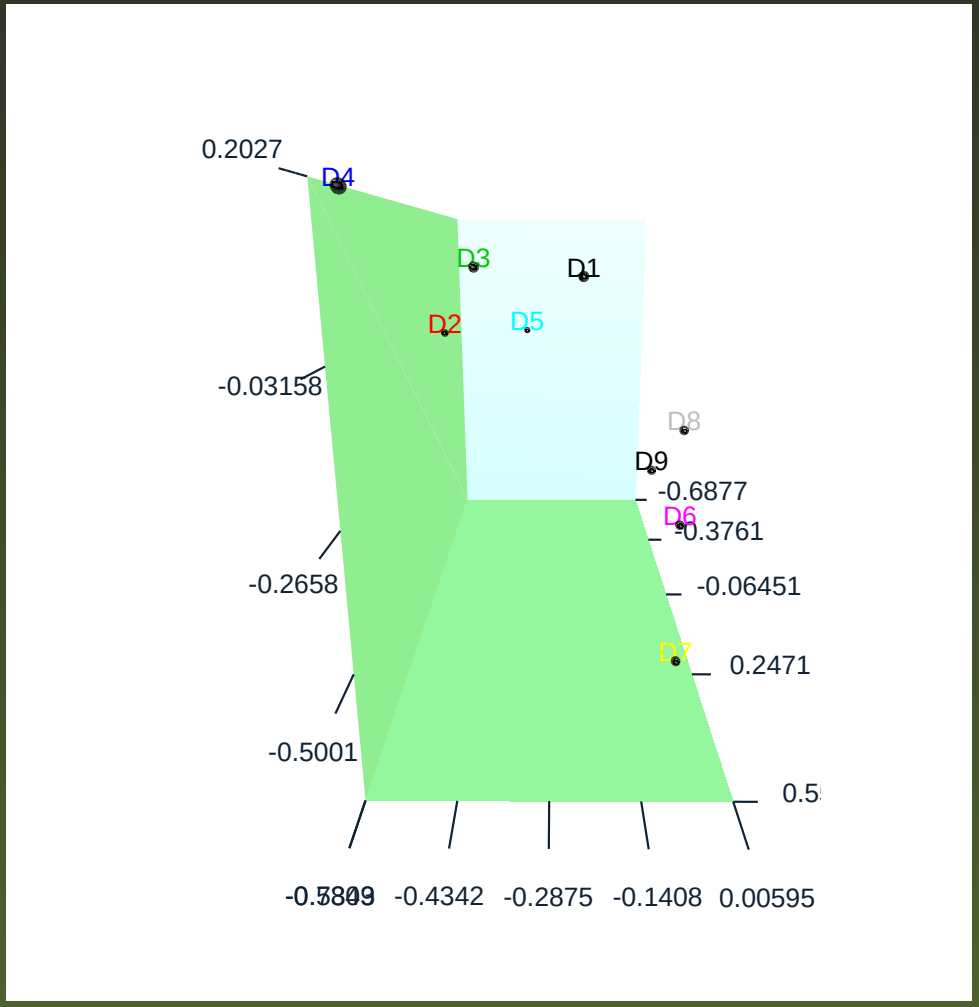
```
# LSA を実行してみる
myLSAspace <-
    lsa(myMatrix,
    dimcalc_share(0.4))
myLSAspace
round(myLSAspace$tk,
    digits= 2)
```

```
# 出力を編集
          [,1]  [,2]  [,3]
abc      -0.06  0.02  0.07
applicat -0.06  0.02  0.07
comput   -0.22 -0.01 -0.04
human    -0.22  0.09  0.26
interfac -0.18  0.05  0.13
machin   -0.06  0.02  0.07
opinion  -0.16 -0.03 -0.11
respons  -0.26 -0.08 -0.35
survey   -0.18 -0.16 -0.09
syst     -0.58  0.13  0.33
```

文書ベクトルの 3 次元表現

```
# 近似文書行列の生成
new3Doc <-
    t(myLSAspace$tk)
    %*% myMatrix
rgl.open()
rgl.bg(color =
    c("white", "black"))
rgl.spheres(new3Doc[1,],
    new3Doc[2,],
    new3Doc[3,])

rgl.texts(new3Doc[1,],
    new3Doc[2,],
    new3Doc[3,],
    rownames(myLSAspace$dk))
```



北 研二 他 (2002)『情報検索アルゴリズム』共立出版

tm パッケージ

- tm_0.2-3 by Ingo Feinerer
  - S4 クラスに基づく実装
  - 各種フォーマット・メタ情報 (HTML,XML,Gmane,RSS) への対応
  - 空白,stopwords の処理 (英独露など 13 言語に対応)
  - stemming の処理 (11 言語に対応)
  - 文章・ターム行列の作成
  - 各種重み付け (tf-idf など)

Feinerer: tmパッケージによる解析例

A.Karatzoglou & I. Feinerer: Text clustring with string kernels in R: Advances in Data Analysis, 2006.

- テキストクラスタリング
  - Reuters-21578 データセットのサブテキスト (1720 文書)
  - bag of words：単語頻度情報
    - 古典的 k-mean 法 (`kmeans()`)
  - String Kernels：文字の位置情報 (`stringdot()`)
    - **kernlab** パッケージによる kernel ベースの技法
    - Kernel k-means (`kkmeans()`)
    - Spectral Clustering (`specc()`)

Feinerer: tmパッケージによる解析例

A.Karatzoglou & I. Feinerer: Text clustring with string kernels in R: Advances in Data Analysis, 2006.

- テキストクラスタリング
  - Reuters-21578 データセットのサブテキスト (1720 文書)
  - bag of words：単語頻度情報
    - 古典的 k-mean 法 (`kmeans()`)
  - String Kernels：文字の位置情報 (`stringdot()`)
    - **kernlab** パッケージによる kernel ベースの技法
    - Kernel k-means (`kkmeans()`)
    - Spectral Clustering (`specc()`)

それぞれが 500 words 前後の文書からなる集合についてカーネル・ストリングの作成に約 2 時間,

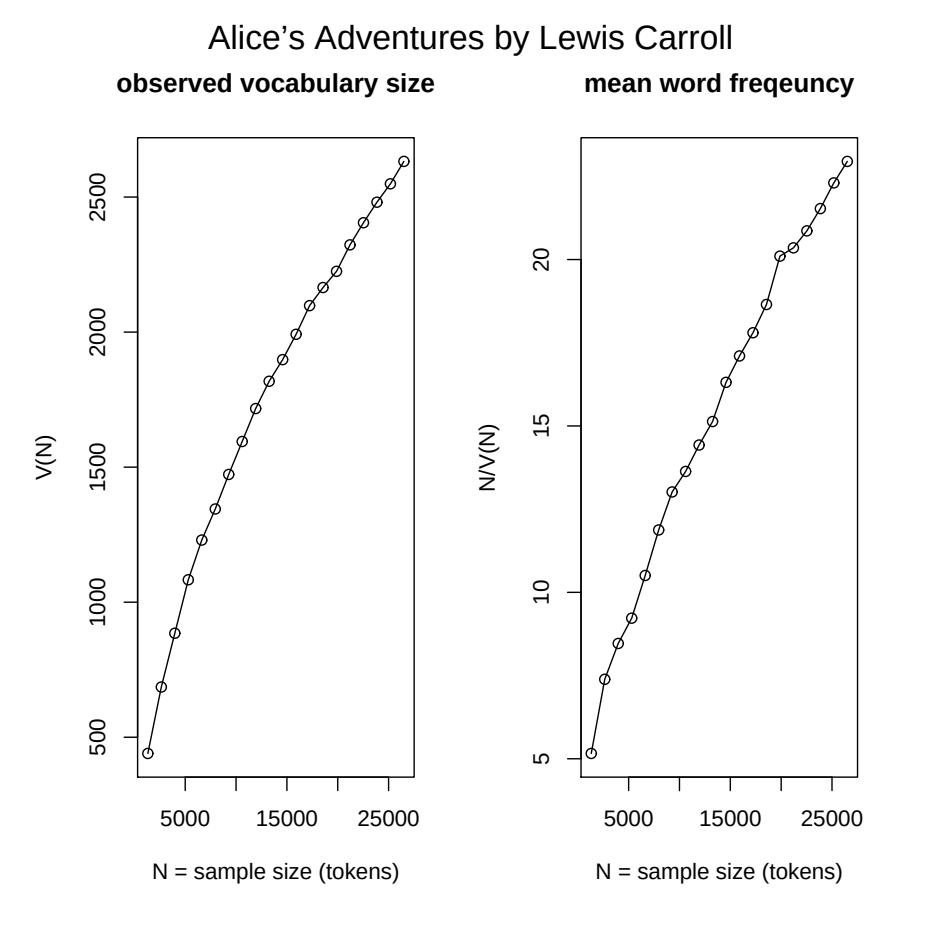
その後のクラスタリングに約 30 分を費やす (2.6 Ghz Pentium 4)

計量言語学: ZipfR パッケージ

例：ボキャブラリの増加率

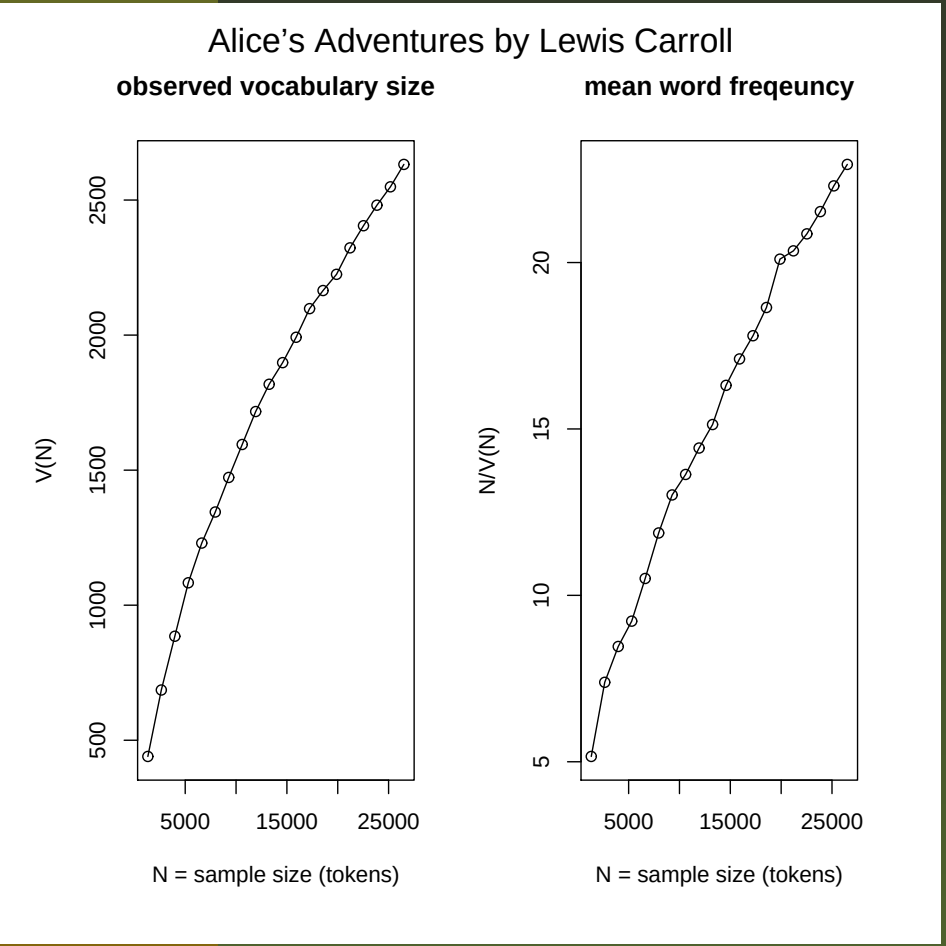
計量言語学: ZipfR パッケージ

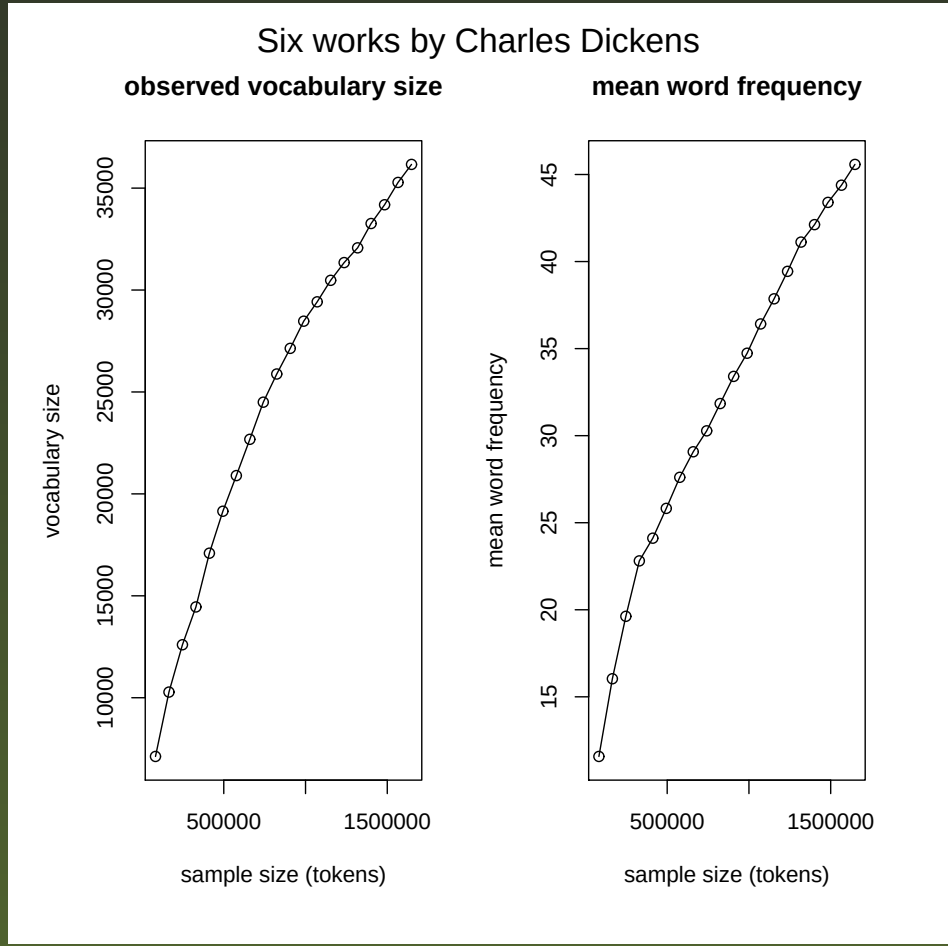
例：ボキャブラリの増加率



計量言語学: ZipfR パッケージ

例：ボキャブラリの増加率





LNRE (Large Number of Rare Events) の問題

参考：Baayen, Word Frequency Distributions, 2002

言語はランダム事象か？

文をランダムに並び替え頻度を測るシミュレーションの例 (実際には 1000 語間隔で計測)

Alice was beginning to get very tired of sitting by (10 語) her sister on the bank, and of having nothing to (20 語) do: once or twice she had peeped into the book (30 語) her sister was reading, but it had no pictures or (40 語) conversations in it, ‘and what is the use of a (50 語) book,’thought Alice ‘without pictures or conversation?’

冒頭から 10 語の範囲で the の頻度は 0

冒頭から 20 語の範囲で the の累積頻度は 1

冒頭から 30 語の範囲で the の累積頻度は 2 … 以下同様

言語はランダム事象か？

文をランダムに並び替え頻度を測るシミュレーションの例 (実際には 1000 語間隔で計測)

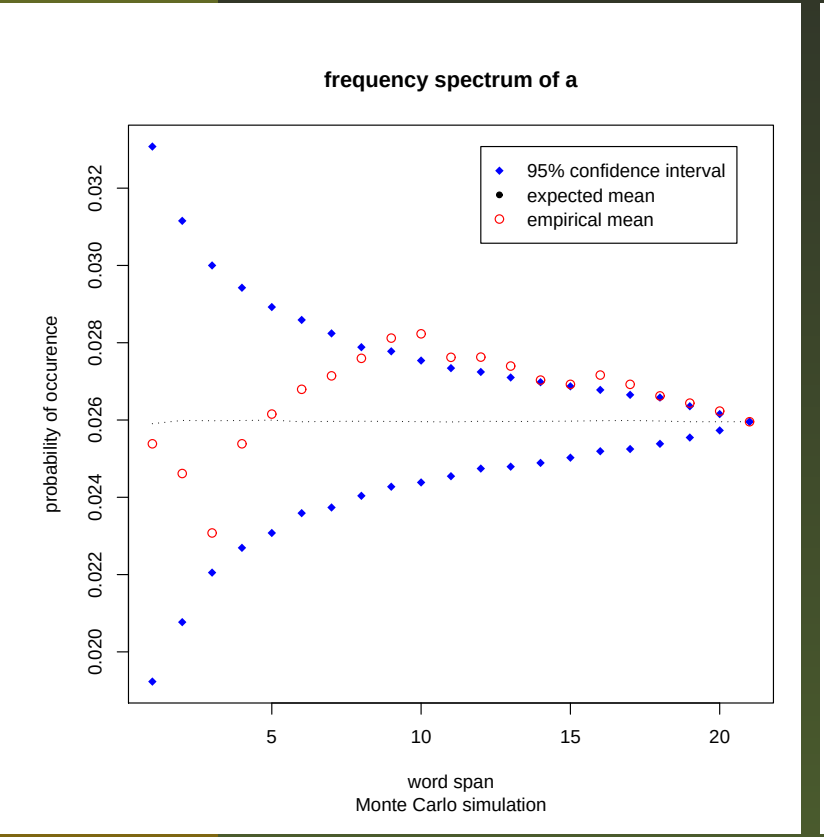
Alice was beginning to get very tired of sitting by (10 語) her sister on the bank, and of having nothing to (20 語) do: once or twice she had peeped into the book (30 語) her sister was reading, but it had no pictures or (40 語) conversations in it, ‘and what is the use of a (50 語) book,’thought Alice ‘without pictures or conversation?’

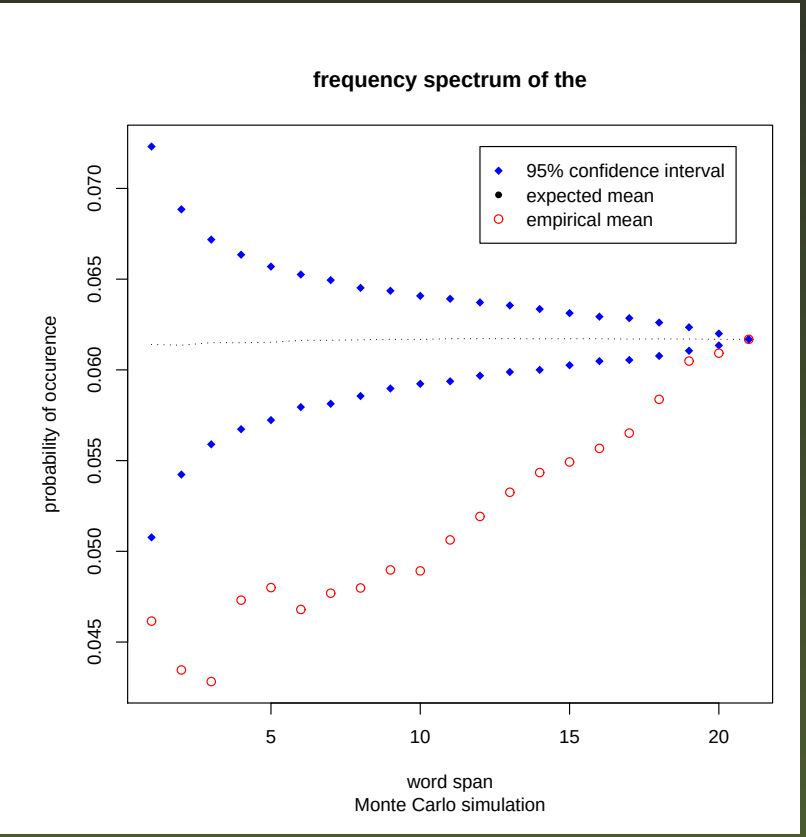
冒頭から 10 語の範囲で the の頻度は 0
冒頭から 20 語の範囲で the の累積頻度は 1
冒頭から 30 語の範囲で the の累積頻度は 2 … 以下同様

what it but to peeped the sister of or was (10 語) was on book no do twice Alice pictures tired or (20 語) of into or by of in beginning the and conversation (30 語) nothing she had thought once very use and reading had (40 語) is without it Alice conversations her a get sister to (50 語) her sitting bank book pictures the having

冒頭から 10 語の範囲で the の頻度は 1
冒頭から 20 語の範囲で the の累積頻度は 1
冒頭から 30 語の範囲で the の累積頻度は 2 … 以下同様

the あるいは a はランダムか





テキストをシャッフルして冒頭から 1000 語間隔で the と a の 頻度を測る．これを 1000 回繰り返す．青いドットはランダムシャフルで求めた 95 %信頼区間．赤いドットはアリスの実測値．参考 Baayen, Word Frequency Distributions, 2002

zipfR パッケージ

- **zipfR** パッケージ, S.Evert & M.Baroni

m	V(m,N)	Sichel	Zipf
1	1176	1176.00	1156.13
2	402	402.64	433.23
3	233	207.35	233.12
4	154	130.25	147.19
5	99	91.18	101.90

- $V(m,N)$ N トークン中 m 回現われた語彙数
- Sichel =Inverse Gauss Poisson model による期待値 $E(V(m,N))$
- Zipf = extended Zipf model による期待値 $E(V(m,N))$

zipfR パッケージ

- **zipfR** パッケージ, S.Evert & M.Baroni
- H. Baayen: Word Frequency Distributions, 2002
- **languageR** パッケージ, H.Baayen

m	V(m,N)	Sichel	Zipf
1	1176	1176.00	1156.13
2	402	402.64	433.23
3	233	207.35	233.12
4	154	130.25	147.19
5	99	91.18	101.90

- $V(m,N)$ N トークン中 m 回現われた語彙数
- Sichel =Inverse Gauss Poisson model による期待値 $E(V(m,N))$
- Zipf = extended Zipf model による期待値 $E(V(m,N))$

文長と確率分布

文長の分布に対して，以下の 4 種の確率分布を検討する

- 対数正規分布 (安本美典『文章心理学入門』1965)

文長と確率分布

文長の分布に対して，以下の 4 種の確率分布を検討する

- 対数正規分布 (安本美典『文章心理学入門』1965)
- ポアソン分布
- 負の二項分布
- Sichel の混合ポアソン分布

文長と確率分布

文長の分布に対して，以下の 4 種の確率分布を検討する

- 対数正規分布 (安本美典『文章心理学入門』1965)
- ポアソン分布
- 負の二項分布
- Sichel の混合ポアソン分布

日本語の場合，文を区切る作業が問題

- 単語と形態素は異なる単位
  (“lik-ed” vs “好き” “だ”)
- 茶筅などの解析ツールを利用

文長解析作業手順

- 青空文庫からサンプリングした 118 の散文
- 茶筅と自作 Java プログラムでテキスト解析
- 形態素 あるいは文字 を単位に文長を計測

文長解析作業手順

- 青空文庫からサンプリングした 118 の散文
- 茶筅と自作 Java プログラムでテキスト解析
- 形態素 あるいは文字 を単位に文長を計測

以下 R による作業

- 118 のテキストごとに頻度の区間幅を段階的に変更
- 作成された頻度表それぞれと四つの確率分布の適合度を測る
- 全テキスト，区間設定，確率分布ごとの結果を集計

負の二項分布

Negative Binomial distribution

$$p_0(x) = \binom{k+x-1}{x} p^k \, q^x$$
$$= \frac{\Gamma(k+x)}{\Gamma(k)x!} \left(\frac{k}{k+\mu}\right)^k \left(\frac{\mu}{k+\mu}\right)^x$$

0-truncated Negative Binomial distribution

$$p(x) = \frac{p_0(x)}{1-p_0(0)} = \frac{p^k}{1-p^k} \binom{k+x-1}{x} p^k \, q^x$$

Sichel の混合ポアソン分布

Sichel's Compound Poisson Distribution

$$\Phi(r) = \frac{((1-\theta)^{\frac{1}{2}})^{\gamma}}{K_{\gamma}(\alpha\,(1-\theta)^{\frac{1}{2}})} \frac{(\alpha\,\theta/2)^{r}}{r!} K_{r+\gamma}(\alpha)$$

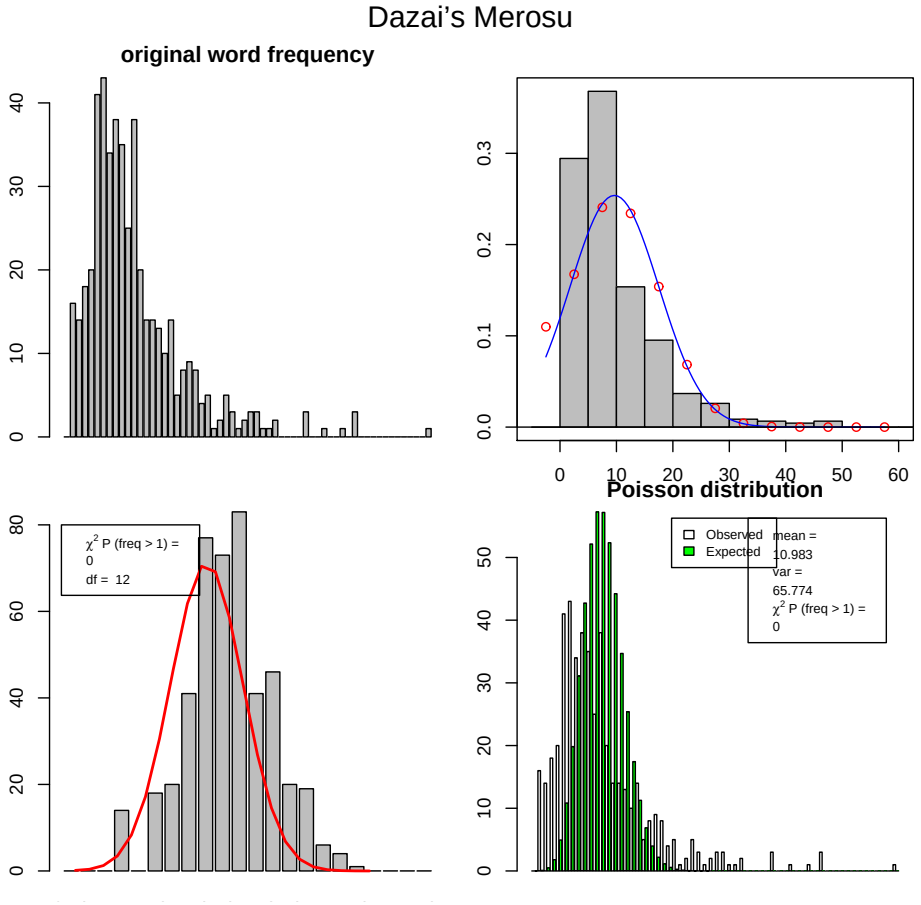
0-truncated Sichel's Compound Poisson Distribution

$$\Phi(r) = [((1-\theta)^{\frac{1}{2}})^{-\gamma}\, K_{\gamma}(\alpha(1-\theta)^{\frac{1}{2}}) - K_{\gamma}(\alpha)]^{-1} \frac{\alpha\,\theta/2}{r!} K_{r+\gamma}(\alpha)$$

ここで $K_{\gamma}()$ は第 2 種の修正ベッセル関数．Sichel によれば $\gamma = -1/2$ が「言語の分布に当てはめがよい」．H. S. Sichel, On a Distribution Representing Sentence-length in written Prose, 1974

走れメロスの分布 – その 1

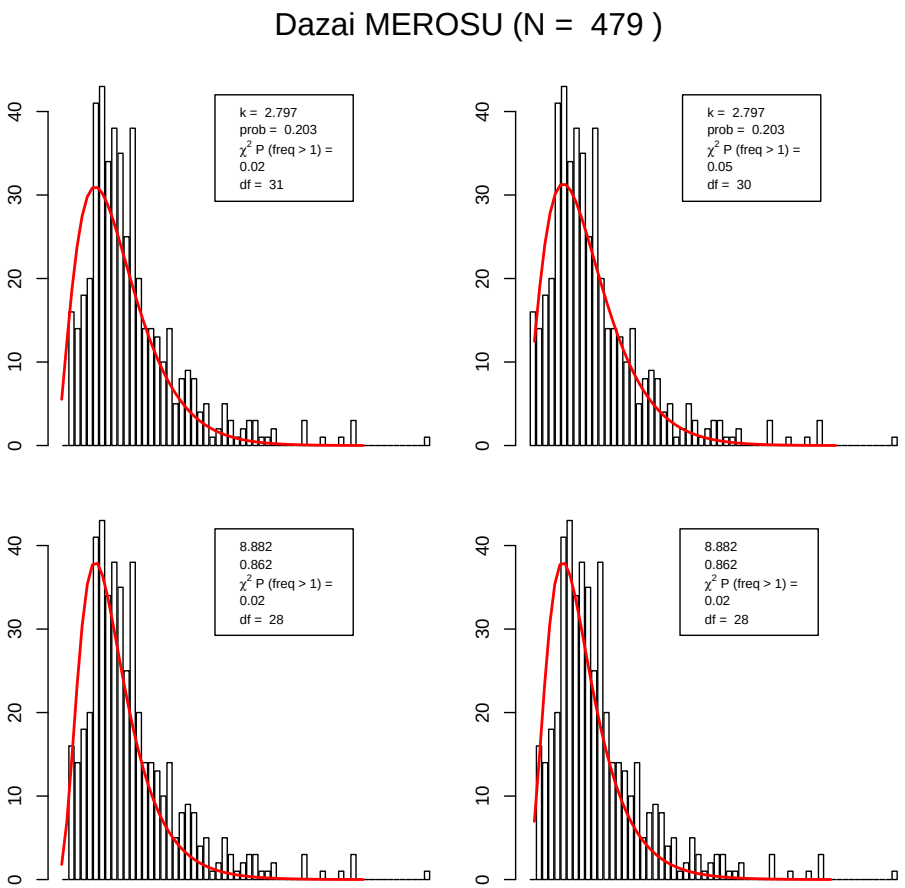
『走れメロス』: 実測値 (上右), 実測値 (上左:確率密度), 対数正規分布 (下左：間隔 0.25), ポアソン分布 (下右)



N	mean	var
479	10.98	65.77

走れメロスの分布 – その 2

『走れメロス』: 負の二項分布，Sichel 分布



確率分布	P
NB	0.02
0-NB	0.05
Sichel	0.02
0-Sichel	0.02

終わり

テキスト解析でも R を使いまわそう !!

終わり

テキスト解析でもRを使いまわそう !!

ご清聴有難うございました.