外国語教育メディア学会 (LET) 関西支部 メソドロジー研究部会 2011 年度報告論集
阪上 辰也 (pp. 8–14)

統計解析環境「R」を利用した言語データの処理

阪上 辰也
広島大学 外国語教育研究センター

概要

本稿の目的は，「R」という統計解析環境を利用した言語データの基本的な処理方法を説明することである。具体的な処理内容として，R を使って言語データ内に含まれる単語数を求めるための基本的な手順を説明する。

Keywords: 統計解析環境 R，言語データ，言語処理

1. はじめに

本稿は，統計解析環境の「R」を利用した言語データの処理方法について説明することである。具体的には，日本人英語学習者の作文データを集めたコーパスである NICE を利用して，基本的なデータの処理方法と，「パッケージ」と呼ばれる付加機能を利用した統計処理方法を紹介する。

2. Rとは何か

R とは，元来，統計処理に特化したプログラミング言語のことを指す。しかしながら，実際には，R のコマンド（何らかの処理を行うためのプログラム）を容易に実行できるよう予めアプリケーションとして動作する形で配布されている。この形で利用できる R は，数値の処理だけでなく，グラフを描いたり，シミュレーションを行ったり，さまざまな処理ができることから，プログラミング言語としてではなく，SPSS や SAS などと同様に「統計ソフト」の一種として認識されている。

R は，無償で利用可能であり，配布サイトから OS に応じてインストール用のファイルを導入し，他のソフトウェアと同様に簡単にインストールできる。インストール後に起動した初期画面は，図 1 のとおりである。インストール方法や基本操作については，多くの Web サイトや書籍で紹介されているため，本稿では割愛するが，R を使用する上で，いくつかの用語を知っておく必要がある。本稿では，変数・関数・作業ディレクトリという 3 つの用語について説明する。

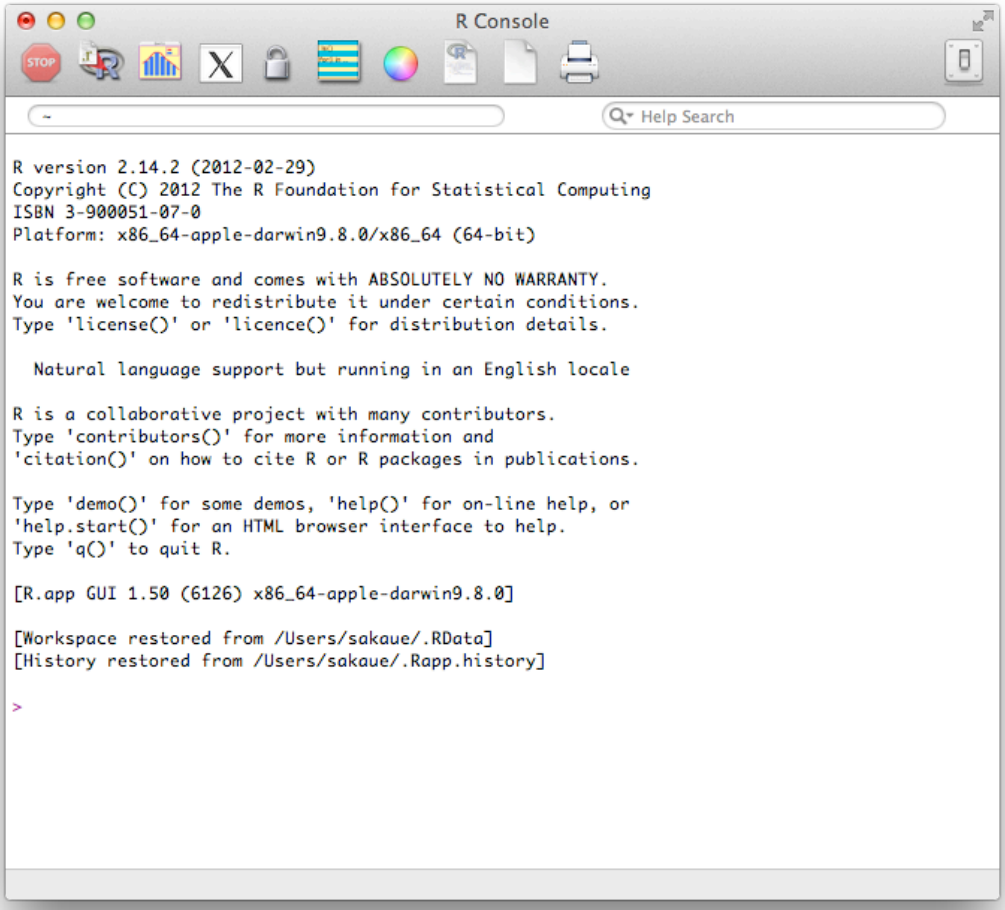

図１．R の起動画面

2.1　変数と関数

R が，元々はプログラミング言語であることを述べたが，R による処理を実行するためには，「変数」と「関数」に対する理解が不可欠となる。

まず，変数は，「複数の値をまとめていれておく箱」として例えられることが多く，数字や文字の書かれた複数のボールが箱の中に入っているような状態であり，この時の箱にあたるものを変数と呼ぶ。「変」という漢字が用いられることからも分かるように，この箱の中身は変化し，10 個のボールが入っていることもあれば，同じ箱でも 100 個のボールが入ることもある。データ処理では，合計値や平均値を求めるなど，複数の値をひとまとまりで扱う必要があるため，変数の利用は不可欠である。

次に，関数とは，「何らかの処理を実行し，その結果を返すもの」として定義することができる。この時，何を対象に処理を実行するのかを指定しなければならないが，その対象となるのは基本的に変数である。関数と変数は，互いに不可欠な存在であり，一対のものとしてとらえる必要があるだろう。

例えば，1 から 5 までの 5 つの数字があり，その 5 つの数値の合計値を求めよという命令を R で実行しようした場合，sum という関数を使用する。この sum 関数を使うと，1 から 5 までの 5 つの数値を足しあわせるという処理を行い，15 という値を返すところまでを行う。以下は，hako という名前をつけた変数の中に，1 から 5 までの 5 つの数値を入れる処理をし，hako という名の変数の中身を確認した上で，合計値を求めるという処理を行なった例である。なお，複数の値をまとめて 1 つの変数の中に入れる場合，c 関数という別の関数を用いる必要がある。

```
> hako <- c(1,2,3,4,5)↵ # c 関数を使い hako という変数に値を 5 つ代入
> hako ↵                 # hako という変数の中身を表示
[1] 1 2 3 4 5
> sum(hako) ↵            # hako の中にある 1 から 5 までの数値を合計する
[1] 15
```

このように，R を利用する際には，変数と関数を頻繁に利用する。「変数に値を入れて（代入する），関数で処理する」という一連の流れを覚える必要がある。

2.2 作業ディレクトリ

作業ディレクトリとは，「データの読み書きを行う場所」のことである。例えば，手元に Excel ファイルとしてしたデータがあり，それを R に読み込ませて処理する場合，作業ディレクトの中に，当該ファイルを置いておく必要がある。また，R で処理したデータを書き出す（保存する）場合，新たに生成されるファイルは作業ディレクト上に保存される。なお，作業ディレクトリがどこに設定されているかを確認するために，下記のように，getwd 関数（Get Working Directory の略）を利用する。

```
> getwd() ↵
[1] "/Users/sakaue
```

3. Rによる言語データ処理の基本

本節では，日本人英語学習者コーパスの NICE を利用して，R によるコーパス分析に際して必要となる基本的なデータ処理手順を説明する。なお，NICE の詳細については阪上他(2008)，また，基本的なデータ処理方法については阪上（2011）を参照されたい。R を利用した基本的なデータ処理の手順を説明する。まず，コーパスデータの処理手順は，以下の 5 段階に分けることができるだろう。

1. データを読み込む
2. データを分解する
3. データを揃える
4. 数値を求める
5. データを保存する

以下では，順を追って，R による基本的なデータ処理方法を説明する。NICE には，習熟度等の書き手の情報が含まれているが，説明をより平易なものとするため，テキスト部分だけを抽出したデータを利用する。

3.1 データを読み込む

R には，多数の関数が用意されており，関数を利用して各種処理を行う。まず，データを読み込むには，scan 関数を利用する。今回利用する学習者の作文データが保存されている nns_raw.txt は，作業ディレクト上に置いておく必要がある。[2]

```
> nns <- scan("nns_raw.txt", what="character")
Read 62959 items
```

上記のように scan 関数を使うことで，空白で区切られたものをひとつずつデータとして読み込まれていく。なお， nns_raw.txt というファイルにあるコーパスデータは，各単語が空白で区切られているため，1 単語ずつ読み込まれるということになる。読み込みが終了すると，「Read 62959 items」という表示が現れ，scan 関数の実行結果として 62959 項目が読み込まれたことを知らせてくれる。

3.2 データを分解する

次に，データを分解する。これは，scan 関数により読み込んだデータが，1 つの「ベクトル」として nns という名前の変数に入っているためである。ベクトルとは，複数のデータを 1 つの要素にまとめたものを指す。したがって，nns という変数には複数の単語データがまとめて入っているが，ベクトルとしては 1 つとなる。言い換えれば，すべての単語が 1 行に並んでいるような状態であり，このままでは単語データの並び替えを行うことができない。そこで，ベクトルとしてまとまっているデータを 1 単語ずつのデータに分解する必要がある。データを分解するためには，strsplit 関数を利用する。

```
> nns_list <- strsplit (nns, " ")
```

この strsplit 関数による処理を行うことで，nns という変数内にある単語データを，空白を区切りとして分割できる。続いて，nns_list の中には，単語が語順どおりに並んだ状態でリストされている。このリストされた状態を解消しない限り，並び替えを行うことができないため，最後に，unlist 関数を使って，単語のリストを分解する。

```
> nns_unlist <- unlist(ns_list)↵
```

3.3 データを揃える

データの分解を終えたら，続いて，データの並び替えなどを行う。この時，主に使用するのは，sort 関数と unique 関数である。3.2 節で単語のリストを分解した状態にあるものを並び替えるため。sort 関数を実行する。

```
> sort_nns <- sort(nns_unlist)↵
```

基本的に，記号類・数値・アルファベットの順でデータが並び替えられる。上記を実行後に sort_nns と入力してエンターキーを押せば，並び替えられたデータを確認することができる。次に，unique 関数を実行して，並び替えたデータを「まとめる」作業を行う。これは，使用された単語のタイプ（異なり語数）を求める際に必要となる。

```
> uniq_nns <- unique(sort_nns)↵
```

これは，使用された単語の総語数（トークン）に加えて，単語のタイプ（異なり語数）を求める際に必要となる。

3.4 数値を求める

まずは，総語数をもとめる。前節で，nns_unlist という変数を作成したが，この変数の中には，すべての単語が分解された状態で入っている。つまり，その変数の中にある要素の数が総語数であり，数値を求めるために length 関数を使用する。結果として，70220 という数値が得られ，この値が（一応の）総語数となる。[1]

```
> length(nns_unlist)↵
[1] 70220
```

続いて，各単語が何度用いられていたかを示す単語頻度一覧表を作成する。この時，度数分布表を作成する table 関数を実行する。

```
> nns_all <- table(nns_unlist)↵
```

さらに，異なり語数を求める場合は，再び length 関数を利用し，変数 uniq_nns の中にあるデータ数を求めればよい。

```
> nns_type <- length(uniq_nns)↵
> nns_type ↵
[1] 7579
```

上記のように，単語の異なり語数は 7579 であったという結果が得られる。つまり，この学習者データの Type Token Ratio は，約 11%（＝7579/70220）ということになる。母語話者のデータに対しても同じ処理を行えば，学習者と母語話者の比較が可能となる。

3.5 データを保存する

結果のデータを新規で保存する場合，write.table 関数を使う。この時，保存したい変数名と，保存するファイル名を入力する。

```
> write.table(nns_all, file="freq.txt" sep="¥t")↵
```

上記の場合，nns_all という変数のデータ，つまり，語彙頻度一覧表のデータが，freq.txt という新たなファイル名で，かつ，タブ区切りのテキスト形式の状態で保存される。あとは，この新たに保存されたファイルを Excel などで再度読み込み，必要に応じてデータの再加工・集計を行えばよい。

4. パッケージを利用したデータ処理

パッケージとは，ある処理・機能に特化したプログラムのことである。R をインストールした時点で，base と呼ばれる基本パッケージが 1000 種類以上利用できるが，さらに特殊な処理を行う場合，新たなパッケージを導入する必要がある。R では，言語データの処理に特化したパッケージが複数公開されており，データ抽出や統計値の計算を効率的に行うことができる。詳細は割愛するが，以下にパッケージ名とマニュアルの URL を示す。

1) tm : http://cran.r-project.org/web/packages/tm/tm.pdf
2) corpora : http://cran.r-project.org/web/packages/corpora/corpora.pdf
3) LanguageR : http://cran.r-project.org/web/packages/languageR/languageR.pdf

なお，上記のパッケージを利用する際は，R 本体のバージョンを必要に迫られない限り更新しないことが重要である。なぜなら，R 本体を更新することで，パッケージが新しい R 本体に対応できず動作不良を起こすことがあるからである。そのため，パッケージの利用に際しては，R 本体とパッケージが安定して動作している状態を保っておくことが重要である。

5. おわりに

本稿では，統計解析環境の R を利用した言語データの処理方法について説明した。具体例として，日本人英語学習者の作文データを集めたコーパスである NICE を対象とし，R の関数や変数を使ってデータを加工し，総語数や異なり語数といった数値を求めるまでの処理過程を述べた。さらに，パッケージと呼ばれる付加機能についても簡単に紹介した。R は，統計ソフトとしても強力なツールであるが，言語処理のためのツールとしても有用であり，利用がさらに進むことを期待したい。

注

1. 記号類が含まれた状態での数値であり，正確な総語数ではない。事前に記号類を削除する処理を施す必要がある。詳細な処理方法は，杉浦(2009)を参照されたい。
2. データの入手先：http://sugiura5.gsid.nagoya-u.ac.jp/nice/

参考文献

阪上辰也 (2011). 「学習者コーパス入門―NICE を利用して―」 『外国行教育メディア学会関西支部メソドロジー研究部会 2010 年度部会報告論集』, 74–99.

阪上辰也・杉浦正利・成田真澄 (2008). 「学習者コーパス「NICE」 の構築」 杉浦正利（代表）『英語学習者のコロケーション知識に関する基礎的研究』平成 17–19 年度科学研究費補助金（基盤研究(B)）研究成果報告書 (課題番号 17320084), 1–13.

杉浦正利 (2009). 「“学習者コーパス論”講義ノート」(アクセス日：2012 年 3 月 28 日) <http://sugiura-ken.org/wiki/wiki.cgi/exp?page=R>