第１回 Ｒプログラミングを始めよう（解答付き）

2023 年 2 月 27 日

目 次

1　ここで学ぶ事

- プログラミングとは?
- コンパイル言語とインタプリタ言語
- R とは?
- R の開発環境
- R Studio の使い方
- 開発手順のおさらい

2　プログラミングとは?

プログラミング
$\Longrightarrow$ コンピュータに処理させたい事柄を，コンピュータが分かる言葉で指示する行為.

2.1　コンパイラ言語

- コンピュータはどんなに複雑な処理も人間が手順を正確に指示すれば，間違いなくこなす事ができる.
  ↓
  アルゴリズム

- コンピュータは電気のオンとオフしか理解できない.
  $\Longrightarrow$ 0（オフ）と 1（オン）で書かれたコンピュータ用の言葉 = 機械語（マシン語）

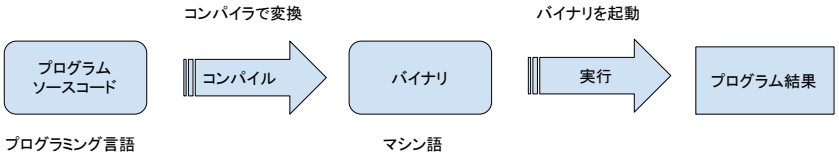
- マシン語（0 と 1 の 2 進数）で書かれたプログラムを**バイナリ（binary）**と呼ぶ.

- 人間が 0 と 1 で書かれたマシン語を操るのは至難の技！人間の言葉に近い言語（= プログラミング言語）からマシン語に翻訳するソフトウェアを使ってプログラミングする．

↓

コンパイラ（compiler）

- プログラミングの手順



- コンパイルしてプログラミングする言語をコンパイラ言語と呼ぶ．

  例: C，C++，Java，Objective-C，Swift，Lisp.
  ただし Java は「仮想マシン」用のバイトコードという言語にコンパイルして実行する．

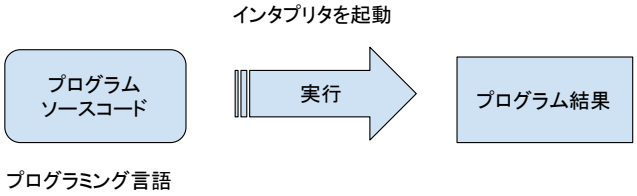
2.2 インタプリタ言語（スクリプト言語）

- プログラムをマシン語に変換して実行するのではなく，プログラムを逐次翻訳しながら実行する方法もある．
  $\Longrightarrow$「インタプリタ」と呼ばれるソフトウェアがプログラムを 1 行ずつ読み込んで解釈し，コンピュータに処理を渡す．

- インタプリタを使って実行するプログラミング言語 ; $\Longrightarrow$ インタプリタ言語
  例) R，JavaScript，Python，Ruby，Perl，PHP，Lisp.

- インタプリタ言語の特徴．
  - コンパイルの必要がない．
  - 書いたプログラムをインタプリタに渡せば実行できる．
  - 実行までの手順が少ないので手軽にプログラミングできる．
  - 同じプログラムを違う OS でも実行できる．
  - コンパイル言語に比べ逐次翻訳作業分だけ実行速度は遅くなる．

- プログラミングの手順



- インタプリタ言語で書かれたプログラム（ソースコード）を特に**スクリプト**と呼ぶ．

2.3 R 言語

- R はインタプリタ言語.
- R 言語でプログラムを書けば，R のインタプリタをインストールしたマシンなら OS を問わず実行できる.
- 様々な OS 用の R インタプリタ（実行環境とも呼ぶ）が無料で提供されている．家のコンピュータにインストールしておこう！
  https://cloud.r-project.org
- 大学のコンピュータ（貸与 PC を含む）でも R が利用できる．起動方法が分からない場合はプログラム相談員に聞くと良い.

2.4 チェックリスト

- プログラミング
- アルゴリズム
- バイナリ
- マシン語
- ソースコード
- コンパイル，コンパイラ
- インタプリタ
- スクリプト

3 R の起動と終了

R の実行環境を起動 $\Longleftarrow$ R 言語のインタプリタを起動していることと同じ.

R を起動すると「>」という記号が現れる．これを入力プロンプトと呼び，ここに R のプログラムを打ち込むと，インタプリタが解釈し実行してくれる.

```
> 1 + 1
[1] 2
> 4 * 3
[1] 12
> 3/2
[1] 1.5
> 10 * 3 + 2
[1] 32
> 10 * (3 + 2)
[1] 50
```

終了するには`quit()`をタイプ．作成したデータなどを保存するか聞かれるので y か n で答える．

- プロンプトにプログラムを 1 行ずつ入力していくと，大きなプログラムを作成するには不便．
  $\Longrightarrow$ テキスト・ファイルにプログラムを書き，それをインタプリタに読み込ませる．

- プログラムが書かれたテキスト・ファイルを**ソースファイル**と呼ぶ．
- 通常，エディタでプログラムを書き，インタプリタを立ち上げてソースコードを読み込む．

3.1 チェックリスト

- R の起動と終了方法
- プロンプト
- ソースファイル

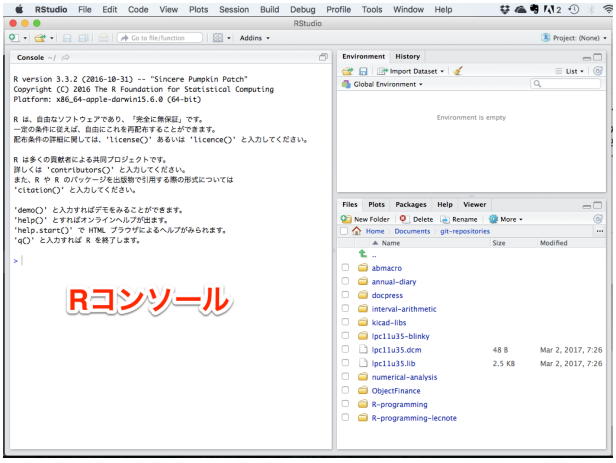
4 統合開発環境—R Studio

前節では R インタプリタを起動してプロンプトにコードを打ち込み，コードを一つずつ実行した．長いプログラムを記述する際は，コードをファイルに記述し，ファイルからプログラムを実行する．ファイルにプログラムを記述するにはエディタが必要である．ここではエディタを含めた開発ツールを 1 つにまとめた RStudio というソフトウェアを使って，ファイルにコードを記述し実行する方法を学ぶ．

- 統合開発環境（Integrated Development Environment）とは，ソースコード用エディタとプログラムの実行環境，デバッガ等を一つにまとめたソフトウェア．
  $\Longrightarrow$ RStudio—無料で使える R 用 IDE （https://www.rstudio.com）
  上のサイトから Open Source 版の RStudio Desktop をダウンロードする．

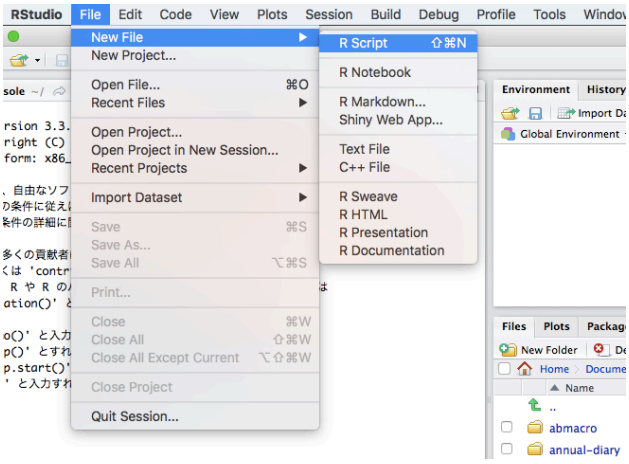
- 起動画面（設定により多少異なる）

  コンソール画面に R のプロンプトが表示される．ここに直接コードを打ち込むことでコードを実行できる．



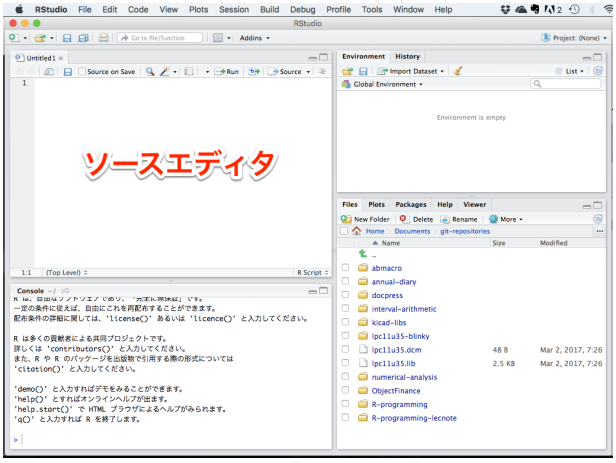
- ソースファイルを新規作成

  コードをファイルに記述するために，ファイルを新規作成する．



- エディタ起動状態

  下の画像では，エディタの起動に伴いコンソール画面が左下に移動している．



- 貸与 PC の場合，作成したソースファイルは必ず USB に保存しておきましょう！

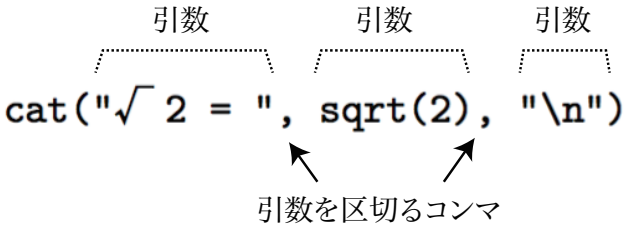
5 開発手順

$\sqrt{2}$ を計算し画面に表示するプログラムの作成．

(1) コンソールでプログラムのプロトタイプ（試作）を作成．
根号記号は日本語入力で「るーと」と打ち込み変換．「\」はバックスラッシュで，Windows ではキーボードの ¥ で代替，Mac では「Option」＋「¥」で入力できる．

```
> cat("√ 2 = ", sqrt(2), "\n")
√ 2 =  1.414214
```

上のプログラムを詳しく見ていこう．cat() は括弧内の項目を画面に表示する関数である．「関数」とはプログラムの処理を 1 つにまとめたもので，入力が与えられると何らかの処理を行い結果を返す．関数への入力は括弧内にコンマで区切って与える．関数に渡す入力を特に「引数（ひきすう）」と呼ぶ．上のプログラムでは cat() 関数に 3 つの引数を渡している．



cat() はこれら 3 つの引数を順に画面に表示する関数である．ダブルクォーテーションで囲まれた第 1 引数と第 3 引数は「文字列」を表す．コンピュータは文字列をテキストデータとして認識する．従って数字もダブルクォーテーションで囲むと，コンピュータは数字として

ではなく文字として認識し，`"1" + 1` を計算することはできない．文字と数字は足しあわせ出来ないからだ．

`cat()` 関数は文字列をそのまま画面に表示する機能を提供する関数だ．ただし，第 3 引数内のバックスラッシュはコントロールシーケンス（制御文字）と言って，次の 1 文字（この場合「n」）を特別な制御文字として扱うようにコンピュータ（この場合 R）に指示する．「n」は newline，すなわち改行を意味し，「`\n`」で画面に改行が挿入される．

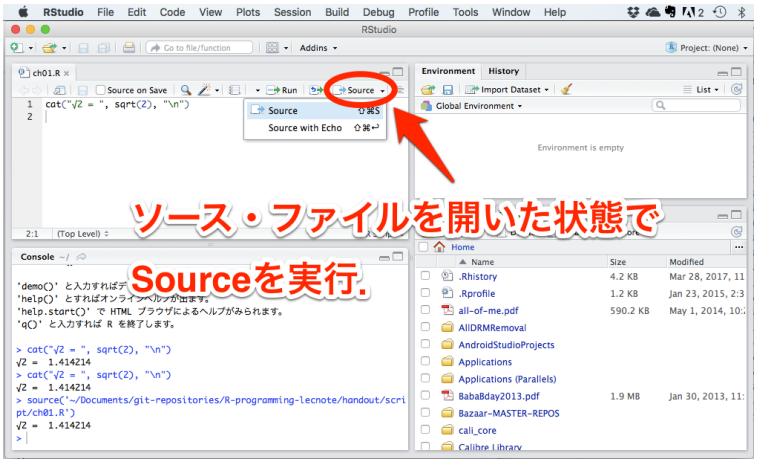
中央の引数，「`sqrt()`」は平方根を求める関数である．

(2) スクリプト・ファイルの作成．“ch01.R” というファイルにプログラムを記述し保存する．

「File」→「New File」→「R Script」からソース・ファイルを作成する．エディタ画面で上のコードを打ち込み保存する．文字エンコードを聞かれたら「UTF-8」を選択し，ファイルの拡張子は「.R」にする．

(3) スクリプト・ファイルからプログラムを実行する．

実行結果はコンソール画面に表示される．



OS によってはファイル名やディレクトリ名に日本語が含まれているとエラーになることがある．その時はファイル名やディレクトリ名を半角英字に変更してからエディタで開き直し Source を実行してみよう．

(4) 「Source」の代わりに「Source with Echo」を実行するとファイルの各行をプロンプトに入力した時と同じように，各行の実行結果が行毎に表示される．

(5) ソースファイル上の特定の行だけ実行したい場合は，その行の頭にカーソルを移動して，Source の左にある Run ボタンを押す．

5.1　チェックリスト

- 関数
- 引数

- 文字列
- コントロールシーケンス
- \n
- cat() 関数
- sqrt() 関数

6 実習

画面に以下のメッセージを表示するプログラムを「hello.R」ファイルに作成し，このファイルからプログラムを実行しなさい.

```
========================
     Hello World!
========================
```

6.1 解答例

```
cat("========================\n")
cat("     Hello World!\n")
cat("========================\n")
```

対話環境で上の様に行毎に 3 回 cat() 関数を呼び出す場合，各呼び出し毎に画面に出力され，以下の様になる（RStudio の「Source with Echo」でも同じ）.

```
> cat("========================\n")
========================
> cat("     Hello World!\n")
     Hello World!
> cat("========================\n")
========================
```

しかし，プログラムをファイルに保存し Source して実行した場合は，R への入力は画面に表示されないため所望の結果を得る.

対話環境（コンソールのプロンプト上）でもうまく表示させたいならば，3 行を連結させて以下の様に 1 行で記述すればよい.

```
cat("========================\n     Hello World!\n========================\n")
```