

R入門

第一回 Rの魅力・超入門

横浜国立大学
酒井信介

講義の目指すところ

- 2回の講義でRについて学ぶ
- Rは統計解析環境(統計を使わないならばあまり意味はない)
- Rを使うことが統計を理解する近道になる
- Rの使い方はすぐに習得できる
- 統計の基礎知識は必要
- 統計学者を目指すわけではない
- 目の前にある関心事に適用し、経験を積み重ねればよい
- 2回の講義で終わるはずはなく、とっかかりを与えるのみ

注意!

- 2回しかないなので、大胆に説明を省略している
- この講義をとっかかりにして、今後は自習により理解を深めていって欲しい
- Rに関する情報は、インターネットの検索でほとんどのものは解決できる

Rプログラミングについて

- プログラマーになる必要はない
- FORTRANやCを経験していれば十分
- あとはルールを覚えるのみ
- 一気にすべてを覚える必要はなく、必要に応じて必要なものから覚えていくというスタンス

選択行 or ドラッグした領域の実行は Ctrl + Enter(Windows)
command+Enter(Mac)

エディタ

オブジェクト内容

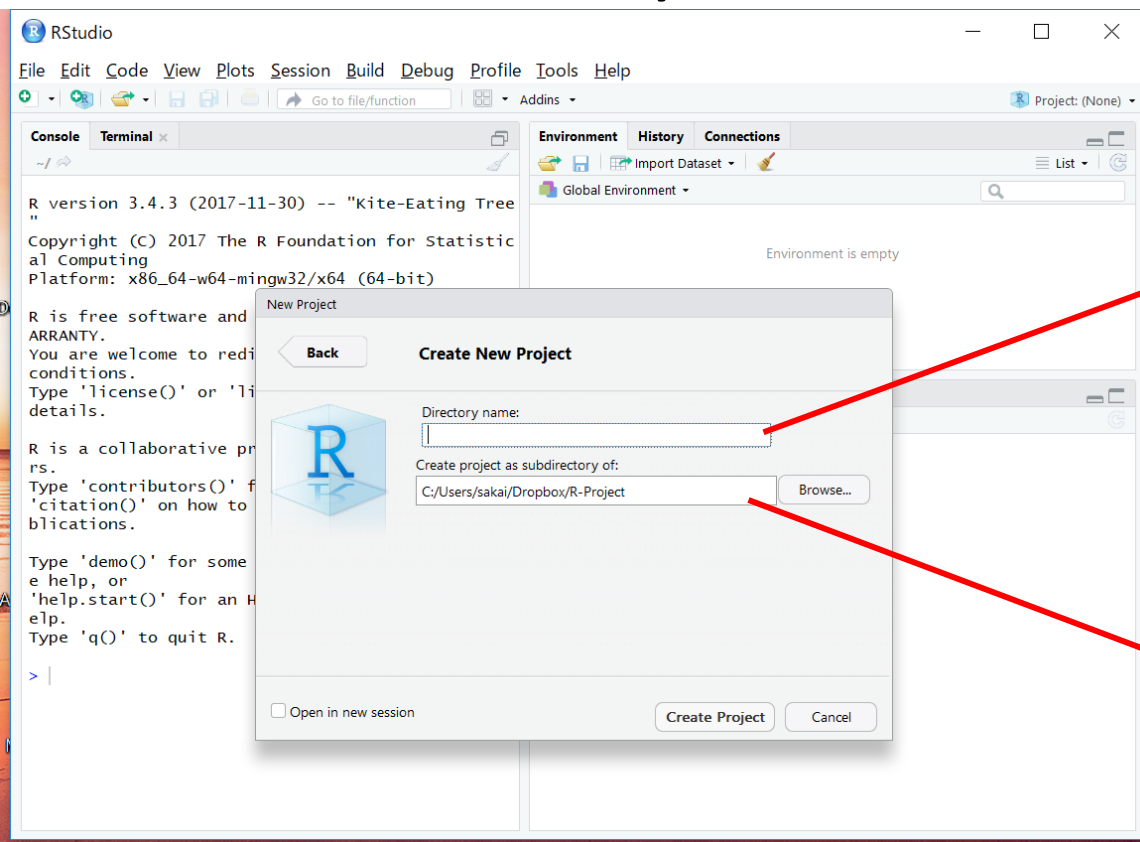
プロット

ファイル表示

コンソール

簡単な例題

- File->New Project
- New Directory



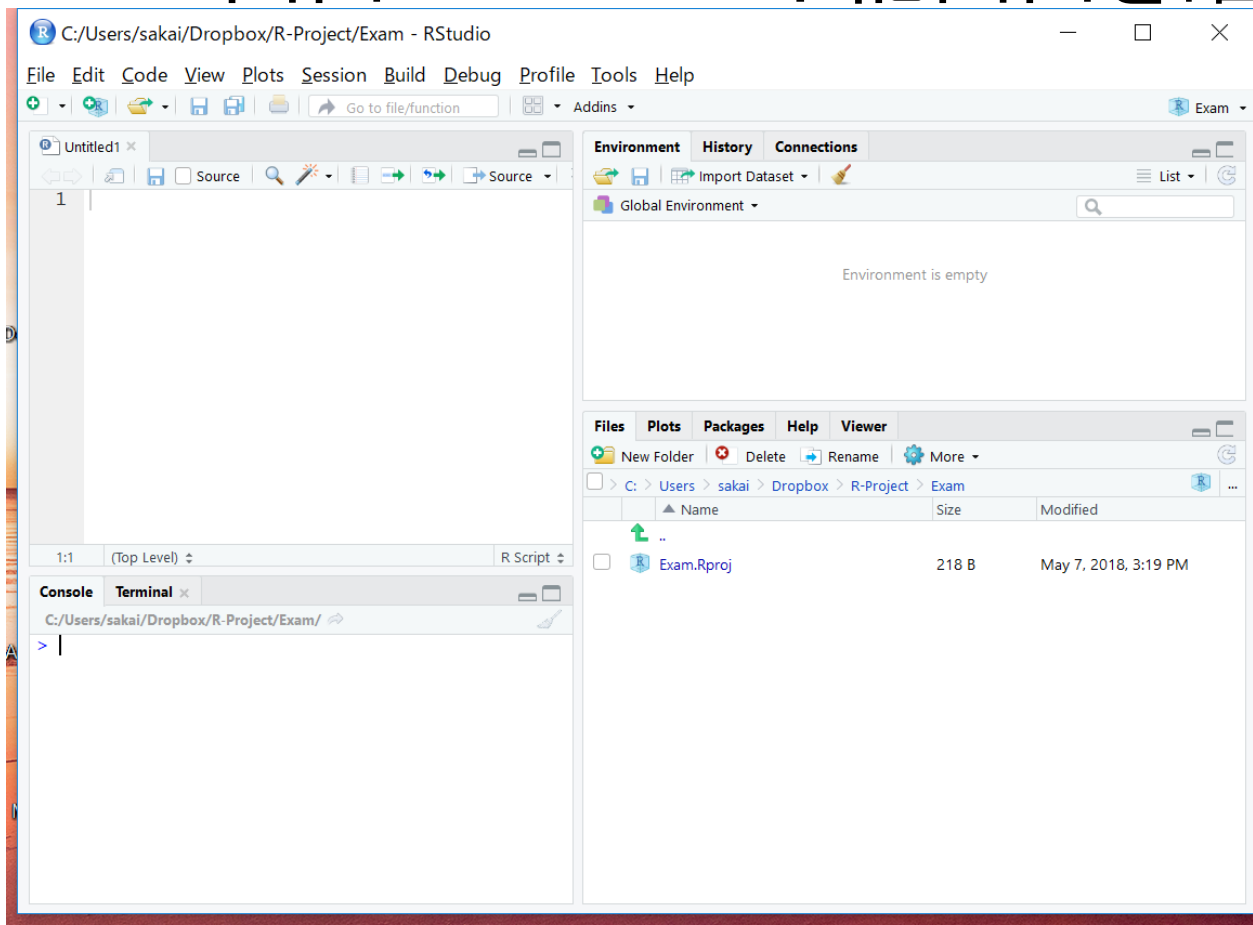
今回作業のsub directory

作業内容を反映した名前とすることが望ましい

Projectを置く親ディレクトリ名

実行方法

- File->New File->R Script
- Console画面でctrl + Iで下記画面を確認



手順

```
print("Hello world!")
```

1. コンソールからの起動
2. エディタ領域に記載してCtrl+Enter
3. File->Save asでセーブした後にSourceボタンを押す

Rを使う前の準備

- R本体だけでも、利用できるが極めて使いにくい
- Rstudioによりはるかに使いやすくなる。ただし、Rを使っているためRのインストールも必要
- Rはコマンドとして実行することもできるし、プログラムとして実行することもできる
- 日本語対応が不十分。最近ようやく実用レベルになったが、使うことは推奨できない

Rの魅力とは(1)

- 他にない高度統計処理機能を有しているにもかかわらずフリーソフトである
- 統計を使うことが求められるが、今から学習することに不安を感じている人には最適
- 統計は、実データに対してさまざまな観点から試行して、初めて現象が見える。本から学ぶ学習では困難。まずは手を動かせ。
- 統計情報は図示されないと理解できないものが多い。Rのグラフィック機能は強力。論文にも使える。

他言語との比較

言語	特徴	価格	大規模データ	計算速度	到達速度
R	統計特化	無料	△ (メモリ依存)	△	◎
SAS/SPSS	統計特化	数十万～	○	○	○
エクセル	高い汎用性	1万～	× (約100万行)	×	×
C++	高速だが多くの労力	無料～数万	実装次第	◎	△

超入門

スクリプトの実行

```
X <- 1  
Y <- 3.4  
Z <- X+Y  
Z
```

この実行によって4.4が出力される。つまり、これまで学んできた言語との対応で考えると<- は= と同じ意味であることが分かる。=と書いても、同じ結果になるが、<-と書くこと。<-と記述することには、特別の意味がある。

R言語は、オブジェクト指向で記載されており、変数や数値は全てオブジェクトである。<-の解釈は、右辺の計算の結果について、左辺のオブジェクトの代入というmethodを実行しているという意味

オブジェクト指向の位置づけ

初めてRを使う人にとって、オブジェクト指向は理解しなくてよいが、この結果として大きな恩恵を受けていることを理解しておくといよい。

コンソールからスクリプトの実行

ls()

“X” “Y” “Z”

現在、システム内にX,Y,Zの3つのオブジェクトが存在している、という意味

変数の型宣言、配列宣言不要

FORTRANやCを学んだ人には、変数の型宣言せずに、いきなりプログラムが始まるのは違和感があるかもしれない

整数 → `X <- 1`
実数 → `Y <- 3.4`
 → `Z <- X+Y`
 `Z`

これらは、全てオブジェクト指向の恩恵(詳細略)

プログラム開発の精神的負担が減るとともに、コンパクトになる

```
a <- rnorm(100)
```

正規乱数100個を、aというオブジェクトに代入。aは自動的に大きさ100の配列オブジェクトとなる。あらかじめ、大きさ100の配列であることを宣言する必要なし

ベクトル計算の活用で効率化できる

例

```
X <- c(3,8,20,16)
```

```
sqrt(X)
```

```
mean(X)
```

```
X*1.1
```

```
Y<-c(1,2,3,4)
```

```
X*Y
```

ベクトル計算の活用で、大幅な計算時間の短縮が実現できる!なおかつプログラムをスマートに書ける

help機能が充実

例

```
X <- c(7,6,30,4,9,2)
sort(X)
sort(X,decreasing=TRUE)
```

sortについて調べたければ

```
help(sort)
```

もしくは

```
?sort
```


Rの魅力とは(2)

- 統計にかかわる高度機能をほとんど持っている。必要に応じてパッケージをインストールする。統計処理のプログラムが容易。
- 利用上の不明点はインターネット情報でほとんど解決できる
- パッケージのマニュアルが充実していて、例題もついているので、実体験による技術習得が容易
- 高度の統計処理が一語のコマンドで処理できてしまうものが多い(postscript出力も可)

Rの魅力とは(3)

- フレームやリスト、マトリックスなど多様なデータ表現が可能。
- 試行するためのサンプルデータが数多く提供されている
- エクセルなどとのデータのやりとりが極めて容易
- オブジェクト指向プログラミングなど高度のプログラミング機能も提供している

欠点も理解しておかなければならない

- フリーソフトであるということは、誰も結果責任を負わないということ(企業によっては利用を認めないところもある)
- 結果の妥当性を判断する目を養う必要あり
- 不可解なことが起きることがあるが、自分で解決することが求められる(本に何も書いていない)。例えば、文字フォントの取扱いなど
- 計算速度は必ずしも早くない。プログラムの書き方で、計算速度はかなり差が出る。

不可解な現象の例

一つのプロジェクトの作業終了後に、続けて別のプロジェクトの作業を継続して行うとする。安易に、このようなことを何度も継続すると、フリーズしてしまうことがある。





発生オブジェクトが、メモリ上で飽和した可能性がある



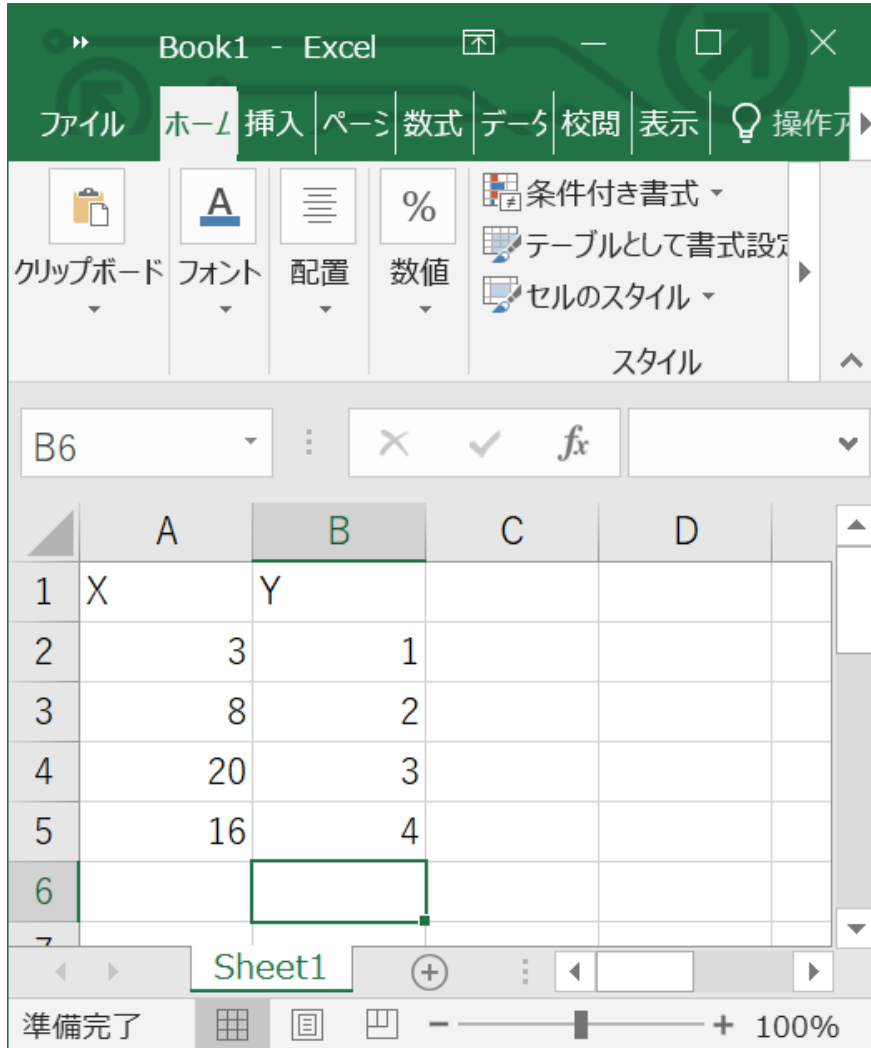
プロジェクト終了後は、そのつどQuitすることが無難

あるいは

`rm(オブジェクト)`  不要オブジェクトの消去

`rm(list=ls())`  全オブジェクトの消去

データの入力



excelでcsvファイルを作ってみる
Project作成中のディレクトリに
exam.csvとして保存

```
aa<-read.csv("exam.csv")
```

aaというオブジェクトにデータが
読み取られる

テーブルデータの参照法

aa\$X → 第一列目

aa\$Y → 第二列目

mean(aa\$X)

は先ほどの例題と同じ結果となることを確認すること