２日目：簡単な計算

本日は，R を使って簡単な計算をしてみます。特に難しいことはないと思いますが…

まず，R を起動しましょう。

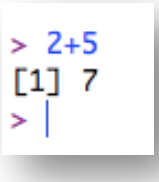
すると，R コンソール内の一番下の方で，カーソルが点滅していると思います。ここが入力するポイントになります。ちなみに「>」は「プロンプト」といい，R が入力を待っている状態を表しています。

四則演算に関する記号は，エクセルと同じです。たし算は「+」，引き算は「-」，かけ算は「*」，割り算は「/」，累乗は「^」もしくは「**」です。すべて半角で入力していきます。また「=」は不要です。

たとえば，「２＋５」を計算させたければ，

2+5

と入力します。そしてリターンを押すと…



こういうふうに結果を返してくれます。なお，[1]という部分は，とりあえず無視しておいてください。それに続いている部分が結果です。

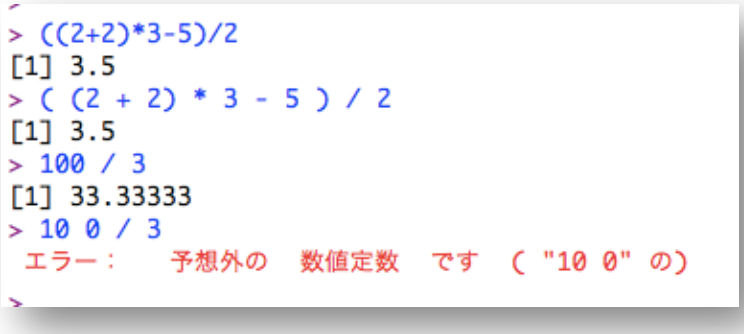
括弧も普通（?）に使えます。

（２＋２）×３　　なら　　(2+2)*3

｛（２＋２）×３−５｝÷２　なら　　((2+2)*3-5)/2

まずは，いろいろと計算を試してみてください。

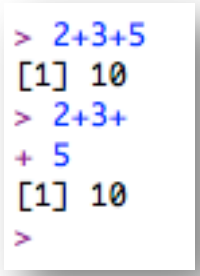
ちなみに，R では半角空白は無視されます（もちろん 100 を 10 0 などと書いてしまうのはダメですが…）。命令をわかりやすく，きれいに書きたい時などにうまく使うとい



いかもしれません。

また，計算式などの命令が途中までの状態（右図でいうと，「2+3+」だけを入力した状態）でリターンを押すと，プロンプト（「>」）の代わりに「+」が表示されます。

そのまま続けて入力すれば，通常通りに計算をしてくれます（逆にいえば，「+」が表示されたということは，命令が完成していないということです）。



もちろん，Rはエクセル同様，ルート，サインやコサイン，logなども計算してくれます。（ここではルートだけ紹介します。他がどのような関数なのかは，webで調べてみてください）

たとえば2のルートなら

```
sqrt(2)
```

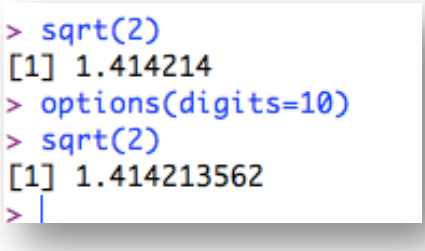
さて，このルート2ですが，「ひとよひとよにひとみごろ」1.41421356…と記憶している人も多いのではないでしょうか。しかし，Rは1.414214という結果を返してきます。これは，Rのデフォルトで最大7桁で表示をすることになっているためだそうです。

これを変更するには，options(digits=)という関数を使います。Rコンソールに以下のように入力し，リターン。この段階では，画面上に特に何の変化もありません。

```
options(digits=10)
```

続いて，再度sqrt(2)と入力すると，今度は1.414213562と返してきます。全部で10桁になっています。digits=は，表示桁数を指示する命令なのです。

なお，このoptions(digits=)という指示の影響は，その後も続きます。もとに戻すなら，options(digits=7)という命令を実行する必要があります。



その計算だけ桁数を変更したいなら，print(計算式 , digits=)を使うとよいと思います。ルート2を10桁で表示させるなら…

```
print(sqrt(2), digits=10)
```

ここではdigits=は省略可なので，print(sqrt(2), 10)でも同じです。

また，どうもdigits=で指定できる最大は22のようです（23以上にするとエラーになる…）。

さて，表示桁数関係でもう一つ。digits=は，全表示桁数を指定するものであり，小数点以下の桁数を指定しているわけではありません。小数点以下の桁数をそろえておきたい場合があります。

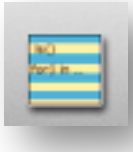
こんな時に使えるのは，round(計算式 , digits = 0)という命令。

ここでの digits =は，小数点以下の桁数を指定します。つまり round(sqrt(2), digits = 0)は，小数点以下が0桁なので，小数第一位で丸めて整数にしなさいということになります。round(sqrt(2), digits = 3)とやっておくと，小数点以下3桁までに丸めてくれます。またこちらも digits=は省略可なので，round(sqrt(2), 3)でも同じです。

なお，ちょっと気をつけておいた方がよいのは，おおむね四捨五入なのですが，厳密に四捨五入をしているわけではないというところです。R は IEEE 式の丸めなのですが，気にしなければならない場合は，詳しいところを調べてください。

ここまでの桁数のコントロールに関する部分をいろいろとやってみると，右のようになります。上で「計算式」と書いた部分は，計算式ではなく，任意の数字でも桁数表示をコントロールしてくれます。

```
> sqrt(2)
[1] 1.414214
> print(sqrt(2), digits=3)
[1] 1.41
> print(sqrt(2), 3)
[1] 1.41
> round(sqrt(2), digits=3)
[1] 1.414
> round(sqrt(2), 3)
[1] 1.414
>
>
> print(12.3456789, digits=3)
[1] 12.3
> round(12.3456789, digits=3)
[1] 12.346
>
```

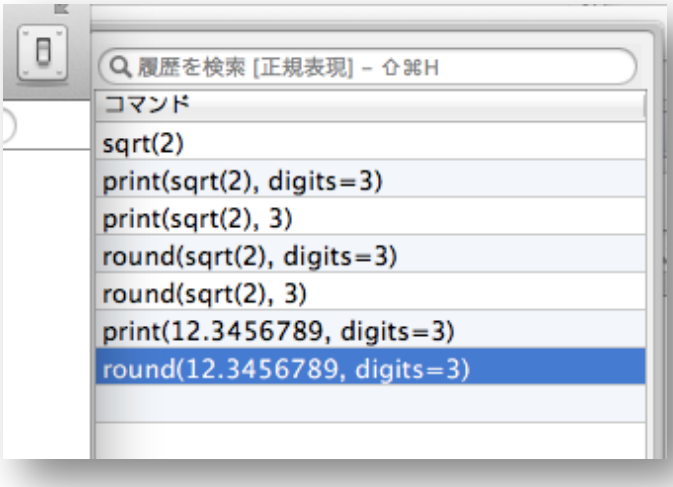


今日は，あと一つ紹介しておきます。R コンソールの上部にあるアイコンから右図のアイコンをクリックしてください。

すると右側にリストが出てきます。

右上図の計算をやらせると，次の図のような記録が残っています。これはRに実行させた命令の履歴です。このようにRは履歴を記憶しています。

履歴にある命令を再度実行させるには，Rコンソールで「上向き矢印キー」を押すと，１つ前，２つ前…の命令を再表示させ，リターン。もしくは，この右側に出てくる履歴一覧から必要なものをダブルクリックするとRコンソールに再表示してくれるので，ここでリターン。これで再計算をしてくれます。この右側に出てくる履歴は，Windowsの Rにはない機能です！



また，履歴の下部にあるように，これを保存しておき，必要な時に呼び出すこともできます。(履歴を保存しようとするとき，拡張子は表示されません。しかし，ファイル名だけを入力しておけば，「.history」という拡張子を自動的につけてくれます。)

２日目は以上で終了です。