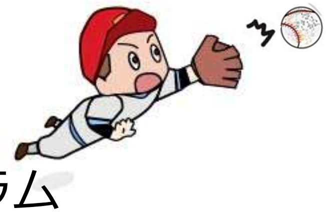
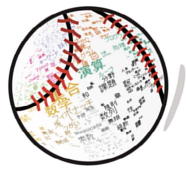
高校生のデータサイエンス・77 本ノック



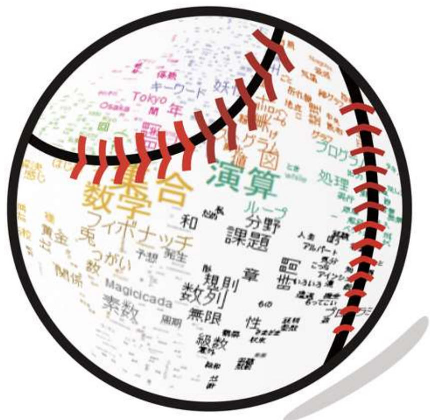
NAIST STELLA プログラム

「共創」が育む主体性の未来　学習教材



付録に「R と RStudio のインストールと使い方」も収録！

金谷重彦 編著



国立大学法人
奈良先端科学技術大学院大学
NARA INSTITUTE of SCIENCE and TECHNOLOGY

ISBN978-4-902874-04-4

高校生のデータサイエンス・77 本ノック

NAIST STELLA プログラム

「共創」が育む主体性の未来 学習教材

付録に「`R` と `RStudio` のインストールと使い方」も収録！

金谷重彦 編著

奈良先端科学技術大学院大学・データ駆動型サイエンス創造センター

先端科学技術研究科・情報科学領域

ISBN978-4-902874-04-4

目次

はじめに

「高校生のデータサイエンス・77 本ノック」は、高校生の方々に「いきなり大学院」（高院連携）、すなわち、ちょっと研究にむけたプログラミングを通した知識発見を実現しようと作成しました。

ここで使っているプログラミングはR 言語で、統計・数理解析をするために洗練された言語です。それと、目的に達するためのプログラム行数が少ないから、比較的理解しやすいという特徴があります。

皆さんは、R 言語との初めての出会いだと思います。巻末には付録として、R のインストールと RStudio のインストールならびに使い方を収録しました。RStudio を使うとプログラミングの作業がとても楽になるので、これを使いこなしましょう。

その上で、「I.プログラミング基礎」と「II.データ整理の基礎」を実習しましょう。その後は、「III 知識発見」のテーマ、3.妖怪、4.関係性とは、5.確率・統計、6.集合、7.フィボナッチ数、8.素数、9.無限級数の和、のどこを読んでもすぐに実習できるようになっています。まあ、興味のありそうなところから読んでください。まあ、内容としては大学院のものもありますが、驚きながら楽しみましょう。

「おっと、面白じゃないか！」
と思っていただきたく、よろしくね！

I. プログラミング基礎

1. R プログラミング基礎 [情報 1]

本テキストで使うプログラム文法を理解しましょう。実際に実行して出力結果により動作原理を解読してください。

変数の定義と代入

数値 1 の変数 a を代入するには、a<-1 とします。この場合、R 言語では、a と a[1]は同じです。ではこのことを確認しましょう。

ノック 1 本目

RK01.R のプログラムを実行し、a と a[1]が同じことを確認しましょう。

RK01.R

```
1 a<-1
2 length(a)
3 a[1]
4 a
```

1 行目：数値 1 を a に代入しています。

2 行目：length()は要素数を数える関数ですので、length(a)を実行すると 1 と出ます。

3 行目：a に数値を代入しているのですが、これは要素数 1 のベクトルとして a に格納されているので a[1]が 1 となります。

4 行目：a の内容を列挙するので 1 となります。

四則演算など

足し算しよう！数値の足し算をする場合、R でも「+」により足し算ができます。

ノック 2 本目

RK02.R を実行し、a+b の結果の c が足し算の結果であることを確認しよう。

RK02.R

```
a<-5
b<-2
a+b
c<-a+b
c
```

演算を楽しもう！

ノック 3 本目

以下の表は、演算記号を説明しています。この表を参考に RK03.R を実行して結果を考察しよう。

記号	演算
a+b	足し算
a-b	引き算
a*b	掛け算
a/b	割り算
a%/%b	割り算（商を整数で出力）
a%%b	割り算の余り
a^b	a^b
factorial(a)	a(a-1)(a-2)…3・2・1 (a の階乗)

RK03.R

```
a<-13
b<-3
a+b
a-b
a*b

a/b
a%/%b
a%%b

a^b
factorial(5)
```

ベクトルの要素ごとの演算

ベクトルとは、a=(1,2,3)、b(4,5,6)のように複数の数値からなる列です。これらの数値をそれぞれベクトルの要素とします。そこで要素ごとに演算についてもノック 3 本目で示した演算記号を活用できます。例えば、

a+b=(1,2,3)+(4,5,6)=(5,7,9)となります。

ノック 4 本目

ノック 3 本目で示した演算記号をベクトルの要素の演算に適用してみよう。RK04.R の実行結果を考察しよう。

RK04.R

```
a<-c(10,20,30)
b<-c(3,2,1)
length(a)
length(b)

a+b
a-b
a*b

a/b
a%/%b
a%%b

a^b
```

条件分岐：if( ){ }else{ }

中央値を求めるプログラムを作ろう！

　　1.1、12.3、8.5、4.2、2.1、9.4、7.0、3.0

と 8 つの数値列が与えられたときに、この中央値はをもとめるにはどうしたらいいだろうか？

まず、数字を小さい順に並べます。

　　1.1、2.1、3.0、4.2、7.0、8.5、9.4、12.3

要素の数は 8 個なので、4 番目と 5 番目の平均値（=(4.2+7.0)/2=5.6 ）が中央値となります。

　では

　　1.1、12.3、8.5、4.2、2.1、9.4、7.0

のときにはというと、

　　1.1、2.1、4.2、7.0、8.5、9.4、12.3

と並べ替えて、要素数が 7 個であるので、4 番目が中央値となります。7.0 となります。ということは、要素数が偶数か奇数かにより、中央値の求め方が計算が異なります。これを if( ){ }else{ }と場合分けをすれば OK となります。

ノック 5 本目

{1.1, 12.3, 8.5, 4.2, 2.1, 9.4, 7.0, 3.0}の中央値を求めるプログラムを作成しよう。

RK05.R

```
x<-c(1.1,12.3,8.5,4.2,2.1,9.4,7,3)

sx<-sort(x)
ns<-length(sx)

if(ns%%2==0){
  medv<-(sx[ns/2]+sx[1+ns/2])/2
}else{
  medv<-sx[1+ns/2]
}

medv
median(x)
```

ループ for( ){ }

a[1]=1, a[2]=2,a[3]=3,a[4] =4 とします。a[1]*a[2]*a[3]*a[4]を計算しましょう。

初期値として mp=1 としましょう。

i の範囲は 1,2,3,4 とします。

i=1 のとき

a[1]をもとに、m*a[1]を計算してこれをまた m という変数に代入します。すると m=1

i=2 のとき、

a[2]をもとに、m*a[2]を計算してこれをまた m という変数に代入します。すると m=1*2=2

i=3 のとき、

a[3]をもとに、m*a[3]を計算してこれをまた m という変数に代入します。すると m=2*3=6

i=4 のとき、

a[4]をもとに、m*a[4]を計算してこれをまた m という変数に代入します。すると m=6*4=24

となり、m=24(=a[1]*a[2]*a[3]*a[4])となりました。

このように、n 個の値が a[1], a[2] …,[n]と n 個の数値が格納されているとき、

for(i in 1:n){ }

とし、i は 1,2,…,n とし、for(i in 1:n){ }の{ }の中で演算をすれば、繰り返し演算ができます。

```
mp<-1
n<-14
for(i in 1:n){
  mp<-mp*a[i]
}
```

ノック 6 本目

RK06.R を実行し動作を理解しよう。

繰り返し演算(ループ処理)を活用し、1 x 2 x 3…x 14 を計算しよう。

RK06.R

```
a<-1:14
mp<-1
for(i in 1:14){
  mp<-mp*a[i]
  print(paste("round[",i,"]=",mp))
}
mp
1*2*3*4*5*6*7*8*9*10*11*12*13*14
```

```
9 factorial(14)
```

1 行目: a[1]=1, a[2]=2, …,a[14]=14 とベクトルを作る。

2 行目: 掛け算の結果を mp とし 1 を代入し初期化する。

3-6 行目：for ループにより、i=1,…,14 として、1-14 の掛け算を行う。

i=1 について、mp x a[1]=1 を mp=1 とする。

i=2 について、mp x 2(=2)を計算し、これを mp に代入する。すなわち、mp=2。

i=3 について、mp x 3(=6)を計算し、これを mp に代入する。

…

i=14 について　mp x 14(=8717829120)を計算し、これを mp に代入し、おしまい。

出力結果を以下に示す。

```
[1] "round[ 1 ]= 1"
[1] "round[ 2 ]= 2"
[1] "round[ 3 ]= 6"
[1] "round[ 4 ]= 24"
[1] "round[ 5 ]= 120"
[1] "round[ 6 ]= 720"
[1] "round[ 7 ]= 5040"
[1] "round[ 8 ]= 40320"
[1] "round[ 9 ]= 362880"
[1] "round[ 10 ]= 3628800"
[1] "round[ 11 ]= 39916800"
[1] "round[ 12 ]= 479001600"
[1] "round[ 13 ]= 6227020800"
[1] "round[ 14 ]= 87178291200"
> mp
[1] 87178291200
> factorial(14)
[1] 87178291200
```

ノック 7 本目

1 x 2 x 3…x 14 を計算しよう。a[1]=1, a[2]=2, …,a[14]=14 とベクトルを作らなくても 14!を計算できます。どうやればいいのでしょうか？for ループを活用して実現しよう。

RK07.R

```
mp<-1
for(i in 1:14){
  mp<-mp*i
}
mp
```

ノック 8 本目

1,2,3,5,8,13 を c(1,2,3,5,8,13)として、これらの総和を行うプログラムを、for ループを活用して作成しよう。

RK08.R

```
ss<-0
for(i in c(1,2,3,5,8,13)){
  ss<-ss+i
  print(paste("SUM",i,"=",ss))
}
```

出力結果

```
[1] "SUM 1 = 1"
[1] "SUM 2 = 3"
[1] "SUM 3 = 6"
[1] "SUM 5 = 11"
[1] "SUM 8 = 19"
[1] "SUM 13 = 32"
```

ループ処理: while( ){ }

1 x 2 x 3…x k について、1,000,000,000 より小さく、最大となる値と、そのときの k を求めてみよう。

while(条件式)について条件式が成り立つ間、while()の内部の演算を行う。ここで注すべきは

　　while(mp<1000000000)

について、i=12 のときに

　　mp(=479001600)<1000000000

であるので、条件を満たし、i=13 のときに mp(=6227020800)となり、

　　mp<1000000000

を満たさず、while()ループをから脱出する。

　そこで、求めたいのは、i=12 のときの mp となる。つまり while ループを抜けたときの値の一個前をどうやったら得られるか？これを考えてみよう。

ノック 9 本目

1 x 2 x 3…x k について、1,000,000,000 より小さく、最大となる値と、そのときの k を求めてみよう。

RK09.R

```
mp<-1
i<-1
while(mp<1000000000){
   mp<-mp*i
   print(paste("round=[",i,"]",mp))
   i<-i+1
 }
mp/(i-1)
i-1
paste("round",i-1,":",mp/(i-1))
```

出力結果

```
[1] "round=[ 1 ] 1"
[1] "round=[ 2 ] 2"
[1] "round=[ 3 ] 6"
[1] "round=[ 4 ] 24"
[1] "round=[ 5 ] 120"
[1] "round=[ 6 ] 720"
[1] "round=[ 7 ] 5040"
[1] "round=[ 8 ] 40320"
[1] "round=[ 9 ] 362880"
[1] "round=[ 10 ] 3628800"
[1] "round=[ 11 ] 39916800"
[1] "round=[ 12 ] 479001600"
[1] "round=[ 13 ] 6227020800"
```

```
> mp/(i-1)
[1] 479001600
> i-1
[1] 13
> paste("round",i-1,":",mp/(i-1))
[1] "round 13 : 479001600"
```

ノック 10 本目

1 x 2 x 3…x k について、1,000,000,000 より小さく、最大の値と、そのときの k を求めてみよう。ノック 9 本目の改良版です。

RK10.R

```
mp<-1
i<-2

while(mp<10000){
  mprev<-mp
  mp<-mp*i
    i<-i+1

    if(mp>10000){
    mp<-mprev
    i<-(i-2)
    break
  }
}
mp
paste("round",i,":",mp)
```

簡単な行列処理

行ごとの集計(演算)では apply(X, 1, F)、列ごとの集計(演算)が、apply(X, 2, F)を使えば簡単にできる。

ノック 11 本目

行が 6 個(r_1,r_2,r_3,r_4,r_5)、列が 3 個(x,y,z)のデータがあるとしよう。

	x	y	z
r_1	1	2	3
r_2	1	3	5
r_3	1	4	7
r_4	1	5	9
r_5	1	6	11

こうゆうデータについて、行ごとの集計(演算)では apply(X, 1, F)、列ごとの集計(演算)が、apply(X, 2, F)を使えば簡単にできる。これを試してみよう。

RK11.R

```
1  x<-c(1,1,1,1,1)
2  y<-c(2,3,4,5,6)
3  z<-c(3,5,7,9,11)
4  
5  dataSet<-data.frame(x,y,z)
6  rownames(dataSet)<-c("r1","r2","r3","r4","r5")
7  dataSet
8  
9  par(mfrow=c(1,2))
10 cmean<-apply(dataSet,1,mean)
11 barplot(cmean,main="apply(dataSet,1,mean)")
12 
13 rmean<-apply(dataSet,2,mean)
14 barplot(rmean,main="apply(dataSet,2,mean)")
```

7 行目：dataSet は

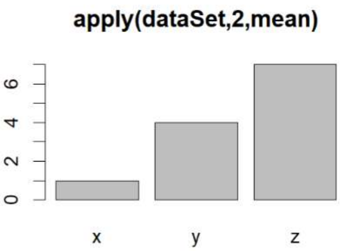
```
> dataSet
   x y  z
r1 1 2  3
r2 1 3  5
r3 1 4  7
r4 1 5  9
r5 1 6 11
```

と定義されている。

9 行目：図を横に 2 枚並べて配置する。

10-11 行目:列ごと(x,y,z)の平均値を求めて、棒グラフに表す（左）。

13-14 行目:行ごと(r_1,r_2,r_3,r_4,r_5)の平均値を求めて、棒グラフに表す（右）。



なお、11 行目の png('RK11.png')から 16 行目 dev.off()までを出力します。ここで、RK11.png が出力ファイル名です。

論理演算 [数学 I]

　論理演算とは、真と偽に関する演算です。「サイコロは六面体だ」を真としよう。一方、「サイコロには 6 の目がある」も真であるとしよう。さらに、「サイコロは五面体だ」を偽としよう。このとき、真を 1、偽を 0 とする。

　『「サイコロは六面体だ」または「サイコロは五面体だ」』は「真」1 + 0=1

　『「サイコロは六面体だ」かつ「サイコロには 6 の目がある」』は「真」1 x 1 =1

　『「サイコロは六面体だ」かつ「サイコロは五面体だ」』は「偽」1 x 0=0

と「かつ」は掛け算、「または」は足し算として真偽を 1,0 により判定できる演算で、ブール代数ともいいます。

ノック 12 本目

{1,5,6,9,8,2,4}それぞれについて、

(1) 2 で割った余りが 0 であることが真(1)か偽(0)か判定するプログラムを作成しよう。

(2) 3 で割った余りが 0 であることが真(1)か偽(0)か判定するプログラムを作成しよう。

(3) (1)を満たす要素を抜き出すプログラムを作成しよう。

(4) (2)を満たす要素を抜き出すプログラムを作成しよう。

(5) (1)かつ(2)を満たす要素を抜き出すプログラムを作成しよう。

(6) (1)または(2)を満たす要素を抜き出すプログラムを作成しよう。

RK12.R

```
1  X<-c(1,5,6,9,8,2,4)
2  TF1<-(X%%2==0)
3  TF2<-(X%%3==0)
4  X[TF1]
5  X[TF2]
6  AND<-TF1&TF2
7  X[AND]
8  OR<-TF1|TF2
9  X[OR]
```

	i	1	2	3	4	5	6	7
1 行目	X[i]	1	5	6	9	8	2	4
2 行目	TF1[i] (X%%2==0)	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
3 行目	TF2[i] (X%%3==0)	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
4 行目	X[TF1]			6		8	2	4
5 行目	X[TF2]			6	9			
6 行目	AND (TF1&TF2)	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
7 行目	X[AND] (残る要素)			6				
8 行目	OR (TF1|TF2)	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
9 行目	X[OR] (残る要素)			6	9	8	2	4

2 行目：「X%%2==0」X[i] (i=1,…,7)を 2 で割った余りが 0 であると TRUE、0 でないと FALSE となります。
4 行目：X[TF1]について、TF1 が TRUE の要素のみを残す。その結果、X[TF1]の要素は、6, 8, 2, 4 となります。
5 行目：X[TF2]について、TF2 が TRUE の要素のみを残す。その結果、X[TF2]の要素は、6,9 となります。
6 行目、**8 行目**: TF1&TF2 とは、「TF1 が TRUE」かつ「TF2 が TRUE」という論理演算です。ここで、積(and)「&」演算と和(or)「|」を以下に示す。参考まで、!X は X の否定です。

X	TRUE	TRUE	FALSE	FALSE
Y	FALSE	TRUE	FALSE	TRUE
X&Y	FALSE	TRUE	FALSE	FALSE
X|Y	TRUE	TRUE	FALSE	TRUE
!X	FALSE	FALSE	TRUE	TRUE
!Y	TRUE	FALSE	TRUE	FALSE

論理演算と四則演算 [数学 I]

論理演算で得られる値には、真(TRUE)と偽(FALSE)があります。ここで「かつ」は&、「または」は|の記号を使います。つまり、TRUE&TRUE は TRUE、TRUE&FALSE は FALSE、FALSE&FALSE は FALSE となります。また、TRUE|TRUE は TRUE、TRUE|FALSE は TRUE、FALSE{FALSE は FALSE となります。R 言語では、TRUE は 1、FALSE は 0 として扱われます。TURE+TRUE は 1+1=2 となります。

ノック 13 本目

RK13.R を実行し TRUE と FALSE の演算「&、 |、 *、 +、 -、 /、 ！」を理解しよう。

RK13.R

```
TRUE&TRUE
TRUE&FALSE
FALSE&FALSE

TRUE|TRUE
TRUE|FALSE
FALSE|FALSE

TRUE*TRUE
TRUE*FALSE
FALSE*FALSE

TRUE+TRUE
TRUE+FALSE
FALSE+FALSE

TRUE-TRUE
TRUE-FALSE
FALSE-FALSE
FALSE-TRUE

TRUE/TRUE
TRUE/FALSE
FALSE/FALSE
FALSE/TRUE

!TRUE
```

論理演算「＊」「|」「！」では、TRUE あるいは FALSE を返す。一方、算術演算では、TRUE は 1、FALSE は 0 として算術演算を行い、数字を返す。

```
> TRUE&TRUE
[1] TRUE
> TRUE&FALSE
[1] FALSE
```

```
> FALSE&FALSE
[1] FALSE
>
> TRUE|TRUE
[1] TRUE
> TRUE|FALSE
[1] TRUE
> FALSE|FALSE
[1] FALSE
>
> TRUE*TRUE
[1] 1
> TRUE*FALSE
[1] 0
> FALSE*FALSE
[1] 0
>
> TRUE+TRUE
[1] 2
> TRUE+FALSE
[1] 1
> FALSE+FALSE
[1] 0
>
> TRUE-TRUE
[1] 0
> TRUE-FALSE
[1] 1
> FALSE-FALSE
[1] 0
> FALSE-TRUE
[1] -1
>
> TRUE/TRUE
[1] 1
> TRUE/FALSE
[1] Inf      (無限大)
> FALSE/FALSE
[1] NaN   (=0/0)
> FALSE/TRUE
[1] 0
>
> !TRUE
[1] FALSE
```

II. データ整理の基礎

2. データの整理 [数学 1]

棒グラフ

barplot()を使うと縦棒、横棒も棒グラフを描画できる。

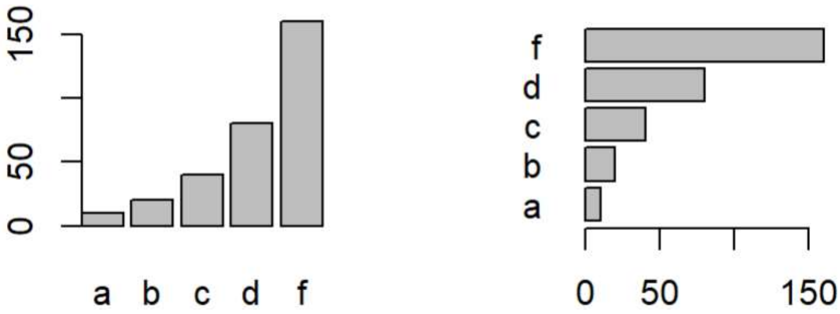
ノック 14 本目

a,b,c,d,f それぞれの値を縦、横の 2 種類の棒グラフを作成しよう。

name	a	b	c	d	f
xdata	10	20	40	80	160

RK14.R

```
graphics.off()
xdata<-c(10,20,40,80,160)
names(xdata)<-c("a","b","c","d","f")

par(mfrow=c(1,2))
barplot(xdata,names=names(xdata))
barplot(xdata,names=names(xdata),horiz=T,las=1,beside=T)
```



barplot()を使うと複数の列からなるデータを棒グラフ行あるいは列で重ねて作成できる。

ノック 15 本目

以下の行列 *dm* について

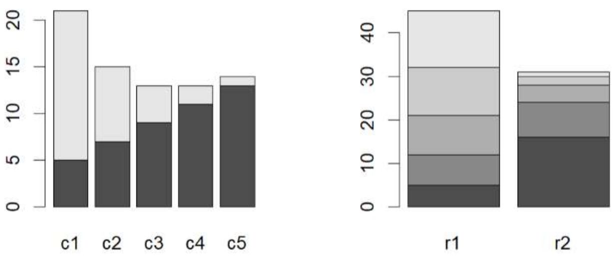
$$dm = \begin{pmatrix} 5 & 7 & 9 & 11 & 13 \\ 16 & 8 & 4 & 2 & 1 \end{pmatrix}$$

行と列についてそれぞれの要素の棒グラフを重ねて描画しよう。

RK15.R

```
graphics.off()
dm=matrix(c(5,16,7,8,9,4,11,2,13,1),nrow=2,ncol=5)
colnames(dm)=c("c1","c2","c3","c4","c5")
```

```
rownames(dm)=c("r1","r2")

par(mfrow=c(1,2))
barplot(dm,names=colnames(dm))
barplot(t(dm),names=rownames(dm))
```



barplot()の変数 biside=TRUE とすると、棒グラフ行あるいは列で重ねずに描画できる。

ノック 16 本目

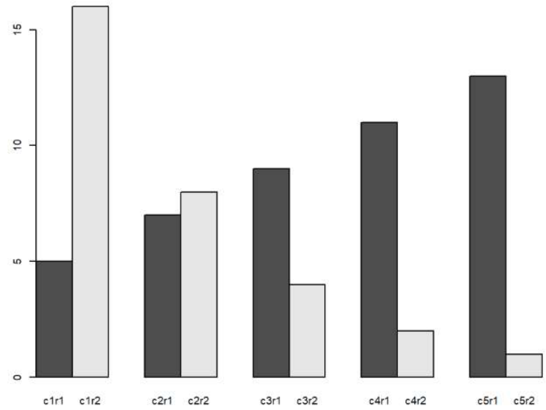
以下の行列 *dm* について

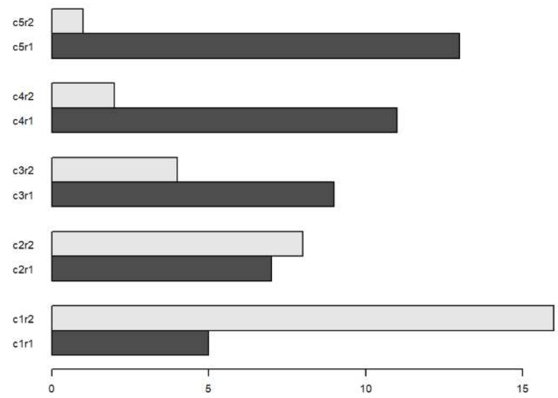
$$dm = \begin{pmatrix} 5 & 7 & 9 & 11 & 13 \\ 16 & 8 & 4 & 2 & 1 \end{pmatrix}$$

行と列についてそれぞれの要素の棒グラフを重ねずに描画しよう。

RK16.R

```
dm=matrix(c(5,16,7,8,9,4,11,2,13,1),nrow=2,ncol=5)
colnames(dm)=c("c1","c2","c3","c4","c5")
rownames(dm)=c("r1","r2")

nr<-dim(dm)[[1]]
nc<-dim(dm)[[2]]
rnames<-rownames(dm)
cnames<-colnames(dm)

nameList<-NULL
for(i in 1:nc){
  for(j in 1:nr){
  pp<-paste0(cnames[i],rnames[j])
  nameList<-c(nameList,pp)
  }
}

par(mfrow=c(1,2),cex=0.6)
barplot(dm,names=nameList,beside=T)
barplot(dm,horiz=T,las=1,names=nameList,beside=T)
```


0
5
10
15
c1r1
c1r2
c2r1
c2r2
c3r1
c3r2
c4r1
c4r2
c5r1
c5r2


c5r2
c5r1
c4r2
c4r1
c3r2
c3r1
c2r2
c2r1
c1r2
c1r1
0
5
10
15

ヒストグラム

hist()を使うとヒストグラムを描画できる。

ノック 17 本目

札幌の 4 月 1-30 日(spDay)の最低気温(spMax)と最高気温(spMin)のデータです。横軸に温度(1 度刻み)をとり spMax と spMin を重ねてヒストグラムで表してみよう。

spDay	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
spMax	21.9	24.5	23.4	26.2	15.3	22.4	21.8	16.8	19.9	19.2	21.9	25.9	20.9	18.8	22.1
spMin	8.3	13.0	8.4	7.9	7.0	3.7	6.1	8.5	8.6	11.9	12.1	14.4	7.0	10.5	6.6

spDay	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
spMax	20.0	15.0	16.0	22.2	26.4	26.0	28.3	18.7	21.3	22.5	25.0	22.0	26.1	25.6	25.7
spMin	10.6	16.6	19.1	20.1	19.8	24.5	12.6	16.4	13.0	13.3	14.1	14.4	17.0	21.3	24.5

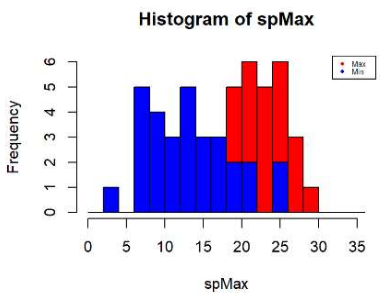
RK17.R

```
1  graphics.off()
2  spDay<-c( 1, 2, 3, 4, 5, 6, 7, 8, 9,10,
3            11,12,13,14,15,16,17,18,19,20,
4            21,22,23,24,25,26,27,28,29,30)
5
6  spMax<-c(21.9,24.5,23.4,26.2,15.3,22.4,21.8,16.8,19.9,19.2,
7            21.9,25.9,20.9,18.8,22.1,20.0,15.0,16.0,22.2,26.4,
8            26.0,28.3,18.7,21.3,22.5,25.0,22.0,26.1,25.6,25.7)
9
10 spMin<-c( 8.3,13.0, 8.4, 7.9, 7.0, 3.7, 6.1, 8.5, 8.6,11.9,
11           12.1,14.4, 7.0,10.5, 6.6,10.6,16.6,19.1,20.1,19.8,
12           24.5,12.6,16.4,13.0,13.3,14.1,14.4,17.0,21.3,24.5)
13
14 hist(spMax,breaks=seq(0,36,2),col="red")
15 hist(spMin,breaks=seq(0,36,2),col="blue",add=T)
16 legend("topright",legend=c("Max", "Min"), col=c("red", "blue"),
17          pch=16,cex=0.5)
```

1 行目：描画のパラメータをいったんクリアします（おまじない！）

2-12 行目：R では spDay<-c(1,2,…,30)のように数値をベクトルで表すときには、c( )の中にカンマ区切って記述します(ファイルからの直接読み込むこともできます、後で説明します)。

14-17 行目：ヒストグラム（頻度表）を描画します）



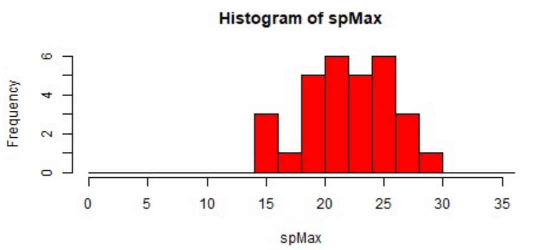
ヒストグラムを重ねないで描画することもできる。

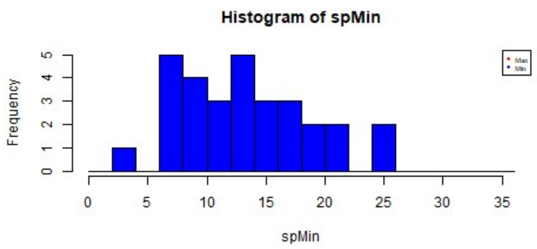
ノック 18 本目

ノック 17 プログラムを 14-17 行目を以下のプログラム 1-5 行と置き換えて実行してみよう。

RK18.R

```
1 par(mfrow=c(2,1))
2 hist(spMax,breaks=seq(0,36,2),col="red")
3 hist(spMin,breaks=seq(0,36,2),col="blue")
4 legend("topright",legend=c("Max", "Min"), col=c("red", "blue"),
5         pch=16,cex=0.5)
```





ヒストグラムと集計

table( )関数では、要素の値あるいは文字列を集計できます。集計結果を棒グラフで表してみましょう。

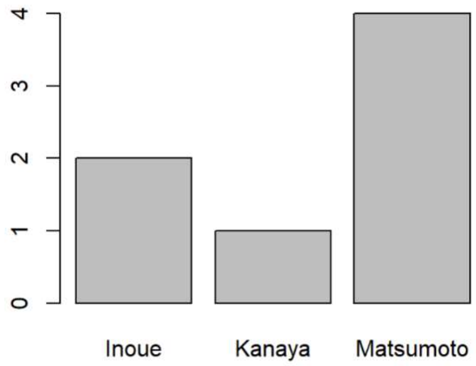
ノック 19 本目

RK19.R では、、table()を活用して、dataC に含まれている 3 種の文字列(Inoue、Matsumoto、Kanaya)を出現数を集計しています。実行して確認しよう。

RK19.R

```
graphics.off()
dataC<-c("Inoue","Inoue","Matsumoto","Matsumoto","Kanaya","Matsumoto","Matsumoto")
table(dataC)

barplot(table(dataC))
```

```
> table(dataC)
dataC
    Inoue    Kanaya Matsumoto 
        2         1         4 
```



ノック 20 本目

RK20.R を実行し、table( )関数と hist( )を比較しよう。

RK20.R

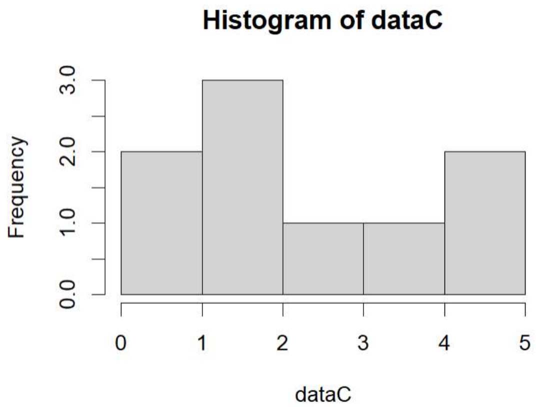
```
graphics.off()

par(mfrow=c(1,2))
dataC<-c(1.5,2,3,4.1,1,3.2,2,1,4.2)
hist(dataC,breaks=seq(0,5,1))

table(dataC)
```

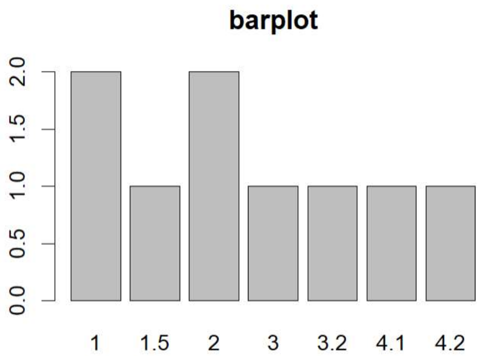
8	barplot(table(dataC),main=”barplot”)

7 行目：table(dataC)として、それぞれの数値の頻度が得られる。

1 1.5 2 3 3.2 4.1 4.2

2 1 2 1 1 1 1

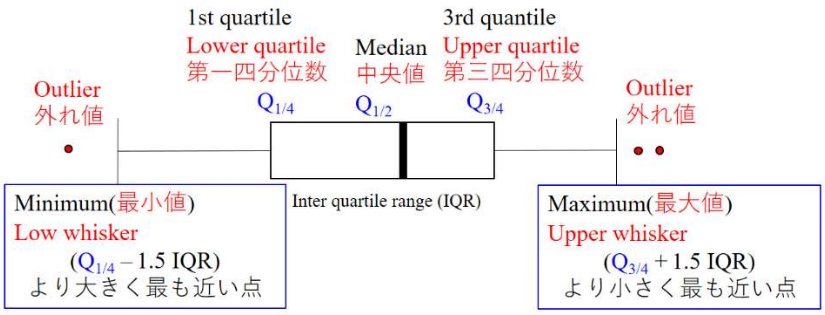




中央値、平均値、箱ひげ図

奇数個の要素からなる集合{1,4,5,8,11}があったとき、これらの中央値は 8 となります。一方、偶数個の要素{1,4,5,6,8,11}があった時、中央は一意に決まらないので、5 と 6 の平均値をとり 5.5 とします。平均値は、数字の合計を要素数で割った値です。つまり、集合{1,4,5,8,11}についての平均値は(1+4+5+8+11)/5 = 5.8 となります。

箱ひげ図とは、以下に示すように、数値を小さい順に並べて、$Q_{1/4}$（第一四分位数；小さい方から 1/4 位の数)、$Q_{1/2}$（中央値；小さい方から 1/2 位の数)、$Q_{3/4}$（第三四分位数；小さい方から 3/4 位の数)をもとに、最小値($Q_{1/4}$-1.5 IQR, IQR=$Q_{3/4}$-$Q_{1/4}$)と最大値($Q_{3/4}$+IQR, IQR=$Q_{3/4}$-$Q_{1/4}$)として、データの分布をみる方法です。



ノック 21 本目

札幌の 4 月 1-30 日(spDay)の最低気温(spMax)と最高気温(spMin)のデータです。spMax と spMin について中央値、平均値を求め、箱ひげ図を描画しよう。

spDay	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
spMax	21.9	24.5	23.4	26.2	15.3	22.4	21.8	16.8	19.9	19.2	21.9	25.9	20.9	18.8	22.1
spMin	8.3	13.0	8.4	7.9	7.0	3.7	6.1	8.5	8.6	11.9	12.1	14.4	7.0	10.5	6.6

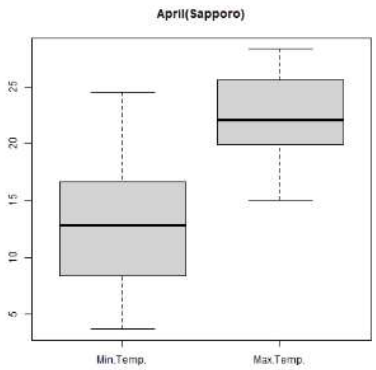
spDay	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
spMax	20.0	15.0	16.0	22.2	26.4	26.0	28.3	18.7	21.3	22.5	25.0	22.0	26.1	25.6	25.7
spMin	10.6	16.6	19.1	20.1	19.8	24.5	12.6	16.4	13.0	13.3	14.1	14.4	17.0	21.3	24.5

RK21.R

```
spDay<-c( 1, 2, 3, 4, 5, 6, 7, 8, 9,10,
          11,12,13,14,15,16,17,18,19,20,
          21,22,23,24,25,26,27,28,29,30)

spMax<-c(21.9,24.5,23.4,26.2,15.3,22.4,21.8,16.8,19.9,19.2,
         21.9,25.9,20.9,18.8,22.1,20.0,15.0,16.0,22.2,26.4,
         26.0,28.3,18.7,21.3,22.5,25.0,22.0,26.1,25.6,25.7)

spMin<-c( 8.3,13.0, 8.4, 7.9, 7.0, 3.7, 6.1, 8.5, 8.6,11.9,
```

```
                  12.1,14.4, 7.0,10.5, 6.6,10.6,16.6,19.1,20.1,19.8,
                  24.5,12.6,16.4,13.0,13.3,14.1,14.4,17.0,21.3,24.5)

mean(spMax)
mean(spMin)

median(spMax)
median(spMin)

graphics.off()
boxplot(spMin,spMax,names=c("Min.Temp.", "Max.Temp."),
            main="April(Sapporo)")
```

```
> mean(spMax)
[1] 22.06
> mean(spMin)
[1] 13.04333
>
> median(spMax)
[1] 22.05
> median(spMin)
[1] 12.8
```



散布図

二つの軸 x と y の値に従ってデータをプロットする図を散布図といいます。

ノック 22 本目

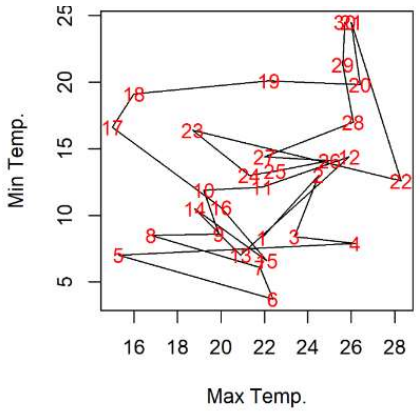
札幌の 4 月 1-30 日(spDay)の最低気温(spMax)と最高気温(spMin)のデータです。spMax と spMin をそれぞれ(x,y)として、1 日目から 30 日までを散布図にプロットしてみよう。

spDay	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
spMax	21.9	24.5	23.4	26.2	15.3	22.4	21.8	16.8	19.9	19.2	21.9	25.9	20.9	18.8	22.1
spMin	8.3	13.0	8.4	7.9	7.0	3.7	6.1	8.5	8.6	11.9	12.1	14.4	7.0	10.5	6.6

spDay	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
spMax	20.0	15.0	16.0	22.2	26.4	26.0	28.3	18.7	21.3	22.5	25.0	22.0	26.1	25.6	25.7
spMin	10.6	16.6	19.1	20.1	19.8	24.5	12.6	16.4	13.0	13.3	14.1	14.4	17.0	21.3	24.5

RK22.R

```
spDay<-c( 1, 2, 3, 4, 5, 6, 7, 8, 9,10,
          11,12,13,14,15,16,17,18,19,20,
          21,22,23,24,25,26,27,28,29,30)

spMax<-c(21.9,24.5,23.4,26.2,15.3,22.4,21.8,16.8,19.9,19.2,
         21.9,25.9,20.9,18.8,22.1,20.0,15.0,16.0,22.2,26.4,
         26.0,28.3,18.7,21.3,22.5,25.0,22.0,26.1,25.6,25.7)

spMin<-c( 8.3,13.0, 8.4, 7.9, 7.0, 3.7, 6.1, 8.5, 8.6,11.9,
          12.1,14.4, 7.0,10.5, 6.6,10.6,16.6,19.1,20.1,19.8,
          24.5,12.6,16.4,13.0,13.3,14.1,14.4,17.0,21.3,24.5)

dataSet<-data.frame(Day=spDay,MaxT=spMax,MinT=spMin)

graphics.off()
plot(dataSet$MaxT,dataSet$MinT,type="l",
     xlab="Max Temp.",ylab="Min Temp.")
text(dataSet$MaxT,dataSet$MinT,labels=dataSet$Day,
     col="red",cex=1,pos=1)
```



折れ線グラフ

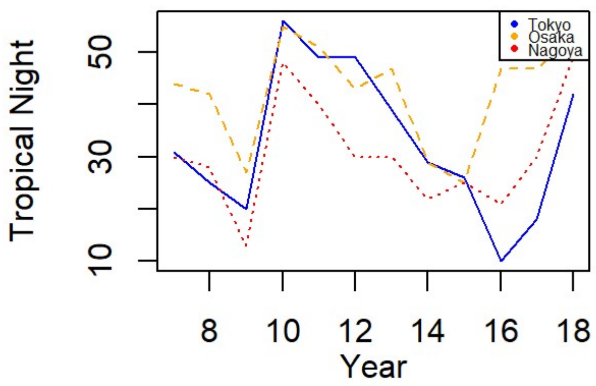
ノック 23 本目

2007 年から 2018 年(Year)のそれぞれの年に、最低気温が 25℃以上（熱帯夜）であった日数を Tokyo、Osaka、Nagoya で集計した。年ごとにどのような変化があるか、3 つの地点で折れ線グラフに表してみよう。

Year	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Tokyo	31	25	20	56	49	49	39	29	26	10	18	42
Osaka	44	42	27	55	51	43	47	29	25	47	47	53
Nagoya	30	28	13	48	40	30	30	22	25	21	30	49

RK23.R

```
Year<-   c( 7, 8, 9,10,11,12,13,14,15,16,17,18)
Tokyo<- c(31,25,20,56,49,49,39,29,26,10,18,42)
Osaka<- c(44,42,27,55,51,43,47,29,25,47,47,53)
Nagoya<-c(30,28,13,48,40,30,30,22,25,21,30,49)

dataSet<-data.frame(Year,Tokyo,Osaka,Nagoya)

graphics.off()
matplot(dataSet[,1],dataSet[,2:4],type='l',
          col=c("blue","orange","red"),xlab="Year",ylab="Tropical Night")
legend("topright", legend=c("Tokyo", "Osaka","Nagoya"),
         col=c("blue","orange","red"),
         pch=16, cex=0.5)
```



III.知識発見

3. 妖怪

NAIST STELLA プログラム「共創」では、「高院の連携実績と広範な学術・国際・地域ネットワークで挑む」、次世代型理数系人材の育成を進めております。その中で、国際日本文化研究センターは、日本文化を国際的な視野で、学際的かつ総合的に研究していこうとする研究機関であり、さまざまな資料を有し、公開を行っています。その中に、

怪異・妖怪画像データベース(DB と略す)(https://www.nichibun.ac.jp/YoukaiGazou/)という妖怪に関する DB を作成し、公開しています。

怪異・妖怪画像 DB の「あ」のところの葵の上(アオイノウエ)をクリックすると、検索結果:1 件と表示され、画像が出力されます(中央の図)さらに、この画像をクリックすると下のほうに、書誌情報が記載されています。葵の上についての資源識別子(U426_nichibunken_0051_0009_0000)がついており、この番号は一つの絵とユニーク(一対一)に対応付けられています。また、主題をみると「鬼：オニ」「鬼女：キジョ」がつけられています。それでは、主題となる用語で、妖怪画がどのくらいあるか、プログラムをつくって検討してみましょう。DataYokai.csv ファイルには、行が資源識別子、列が主題となっています。妖怪の絵（資源識別子）に使われている主題については 1 をそうでなければ 0 となっています。

ID	うさい角(ウサイカク)	うわん(ウワン)	おさん狐(オサンキツネ)	おにぎり(オニギリ)	お化け(オバケ)	お多福(オタフク)	お岩(オイワ)	お歯黒(オハグロ)	お爺さん(オジイサン)	お菊(オキク)	お菊の霊(オキクノレイ)	お菊虫(オキクムシ)	お面(オメン)	かたかい(カタカイ)	かわらけ虫(カワラケムシ)	がま口(ガマグチ)	きこり(キコリ)	くらげ(クラゲ)	ころり(コロリ)	こんがり童子(コンガリドウジ)	ごぼう(ゴボウ)	さら蛇(サラヘビ)	ざんばら髪(ザンバラガミ)	しんかん(シンカン)	すいかん(スイカン)	すりこぎ(スリコギ)	すり鉢(スリバチ)	せむし(セムシ)	ちりとり(チリトリ)	とうもろこし(トウモロコシ)	どうもこうも(ドウモコウモ)	ぬっぺっ坊(ヌッペッポウ)	ぬらりひょん(ヌラリヒョン)	のっぺら坊(ノッペラボウ)	のぶすま(ノブスマ)	はぢっかき(ハヂッカキ)	はら虫(ハラムシ)	ひかん(ヒカン)	ひげ(ヒゲ)	ふさぎ虫(フサギムシ)	ぶらり火(ブラリビ)	ぶんぶく茶釜(ブンブクチャガマ)	ましら(マシラ)	まな板(マナイタ)	もぐら(モグラ)	ももんがあ(モモンガア)	ろくろ首(ロクロクビ)	アシカ(アシカ)	アメノウズメ(アメノウズメ)	ウルユス(ウルユス)	エビス(エビス)	オコゼ(オコゼ)	オサカベ(オサカベ)	オドロシ(オドロシ)	サイコロ(サイコロ)	サザエ(サザエ)	スイカ(スイカ)	タコ(タコ)	ノブスマ(ノブスマ)	一つ目(ヒトツメ)	一つ目小僧(ヒトツメコゾウ)	一本足(イッポンアシ)	一角(イッカク)	丁髷(チョンマゲ)	七福神(シチフクジン)	三つ目(ミツメ)	三つ目入道(ミツメニュウドウ)	三三九度(サンサンクド)	三味線(シャミセン)	三方(サンボウ)	三猿(サンザル)	下駄(ゲタ)	不動明王(フドウミョウ)	不動明王(フドウミョウオウ)	乙姫(オトヒメ)	九尾(キュウビ)	九尾狐(キュウビノキツネ)	九頭竜(クズリュウ)	九頭竜権現(クズリュウゴンゲン)	九頭龍(クズリュウ)
A5_hermitage_0001_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0001_0002_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0001_0003_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0003_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0005_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0006_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0007_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0007_0002_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0007_0003_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0008_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
A5_hermitage_0008_0002_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0008_0003_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0008_0004_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0008_0005_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_hermitage_0008_0006_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_pushkin_0001_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_pushkin_0003_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_pushkin_0004_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_pushkin_0005_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5_pushkin_0006_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ノック 24 本目

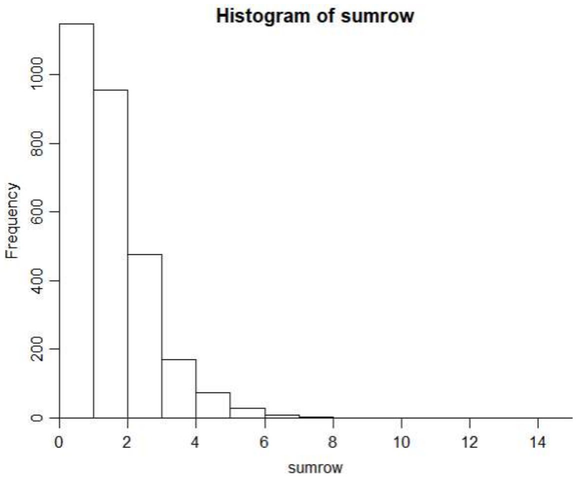
DataYokai.csv をもとに、主題キーワードの個数と、妖怪の絵の数(サンプル数)を求めてみましょう。妖怪の絵にはそれぞれ、主題となるキーワードが使われていれますが、一つの妖怪の絵あたりにどのくらい使われいるか検討してみましょう。また最大の主題キーワード数となっている妖怪の絵を探してみよう。

RK24.R

```
dataYokai<-read.csv("DataYokai.csv",header=TRUE,row.names=1, fileEncoding = "shift-jis")
dim(dataYokai)

sumrow<-apply(dataYokai,1,sum)
graphics.off()
hist(sumrow,breaks=seq(0,15,1))

max(sumrow)
names(sumrow[max(sumrow)==sumrow])
```

> dim(dataYokai)
[1] 2861　872
全ての妖怪の絵は 2861 枚、使われている主題キーワードは 872 個ということがわかる。
> sumrow<-apply(dataYokai,1,sum)
行(妖怪の絵)ごとの使われている主題キーワードの数を数えている。
> hisdata<-hist(sumrow,breaks=seq(0,15,1))

…



ヒストグラムを書くと大半の妖怪の絵には 2-3 個の主題キーワードが活用されいる。
> max(sumrow)
[1] 10
> names(sumrow[max(sumrow)==sumrow])
[1] "U426_nichibunken_0104_0006_0000"
主題キーワードが最も多く使われいる絵の資源識別子は、"U426_nichibunken_0104_0006_0000"であり、なんと 10 個の主題キーワードが活用されている。
　U426_nichibunken_0104_0006_0000 を Google 検索してみると、この絵には、「乙姫；オトヒメ」, 「女；オンナ」, 「魚；サカナ」, 「河豚；フグ」, 「蛸；タコ」, 「女官；ニョカン」, 「冠；カンムリ」, 「扇子；センス」, 「烏帽子；エボシ」,

「食べ物；タベモノ」と 10 個の主題キーワードがついている。

国際日本文化研究センター
怪異・妖怪画像データベース

検索

乙姫;オトヒメ, 女;オンナ, 魚;サカナ, 河豚;フグ, 蛸;タコ, 女官;ニョカン, 冠;カンムリ, 扇子;センス, 烏帽子;エボシ, 食べ物;タベモノ

J.Dautremer

タイトル	
著作者	J.Dautremer
主題	乙姫;オトヒメ, 女;オンナ, 魚;サカナ, 河豚;フグ, 蛸;タコ, 女官;ニョカン, 冠;カンムリ, 扇子;センス, 烏帽子;エボシ, 食べ物;タベモノ
内容記述	乙姫は女房装束に冠を着け、片手に扇子を持って座り、浦島太郎に話しかけている。浦島太郎は公家装束に烏帽子をかぶって座り、盃を片手に乙姫の話を聞いている。彼の前に食べ物が供されている。白い着物に赤い袴という女官のような装束の魚たちが器を運んでいる。そのうちの1体は河豚のようである。同じ装束の魚2体が浦島太郎の脇に座って控えている。公家装束に烏帽子の魚1体と蛸2体も同席している。蛸の1体は片手に扇子を持っている。
公開者	所蔵者:国際日本文化研究センター
寄与者	
日付	2006
資源タイプ	画像
フォーマット	
資源識別子	U426_nichibunken_0104_0006_0000
情報源	親書誌:U426_nichibunken_0104:Ourasima
言語	フランス語

ノック 25 本目

主題キーワードの使用頻度から、妖怪画について検討してみよう。 DataYokai.csv をもとに、主題キーワードの使用回数の表と棒グラフで表してみよう。

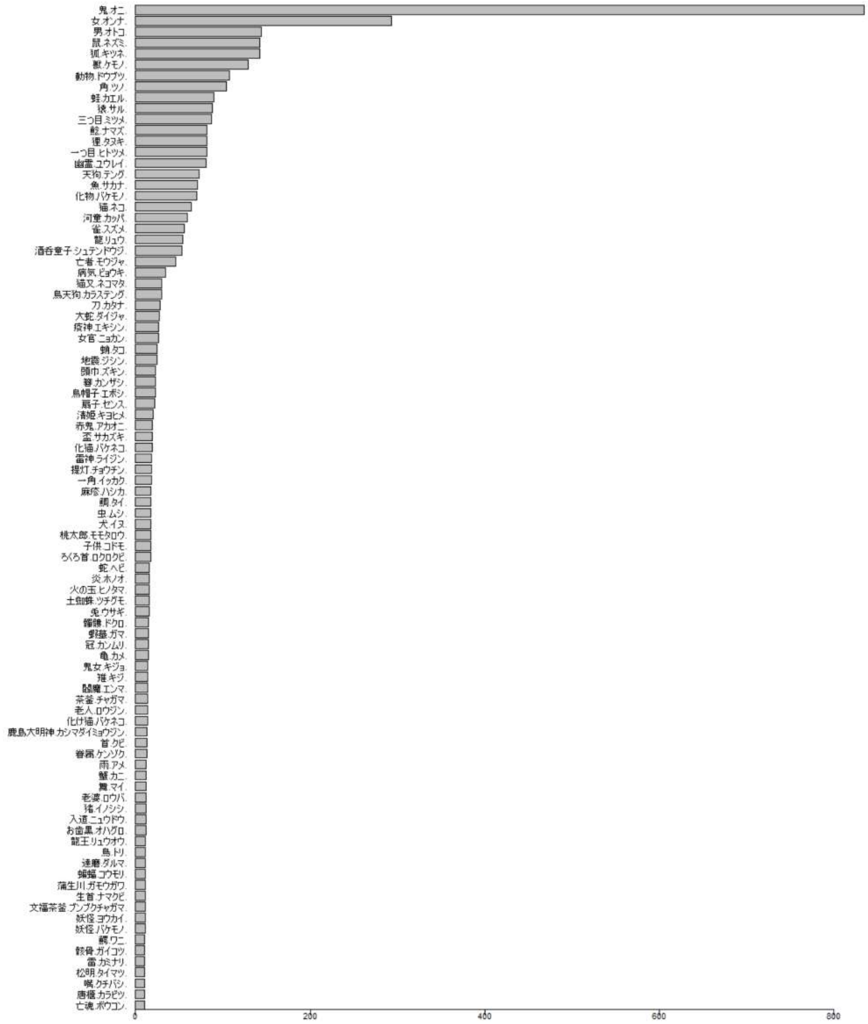
RK25.R

```
dataYokai<-read.csv("DataYokai.csv",header=TRUE,row.names=1, fileEncoding = "shift-jis")

Yokailabel<-apply(dataYokai,2,sum)
sYokai<-sort(Yokailabel,decreasing=FALSE)
s20<-sYokai[sYokai>9]
dataSS<-sort(s20,decreasing=TRUE)
data.frame(names(dataSS),dataSS)

par(mar=c(5,15,1,1),mgp=c(0.5, 0.1, 0),cex=0.2)
barplot(s20,horiz=T,las=1,beside=T)
```

ここでは、10 種以上の妖怪画像で活用されている主題キーワードを以下に示す。主題キーワードの総数 872 個のうち、10 種以上の妖怪画像で活用されている主題キーワードは 92 個だった。ほとんどの主題キーワードは数枚の妖怪画像の説明に使われている。

```
> data.frame(names(dataSS),dataSS)
                              dataSS
鬼.オニ.                         835
女.オンナ.                       293
男.オトコ.                       144
狐.キツネ.                       142
鼠.ネズミ.                       142
獣.ケモノ.                       129
動物.ドウブツ.                   108
角.ツノ.                         104
蛙.カエル.                        90
猿.サル.                          88
三つ目.ミツメ.                    87
一つ目.ヒトツメ.                  82
狸.タヌキ.                        82
鯰.ナマズ.                        82
幽霊.ユウレイ.                    81
天狗.テング.                      73
魚.サカナ.                        71
化物.バケモノ.                    70
猫.ネコ.                          64
河童.カッパ.                      59
雀.スズメ.                        56
龍.リュウ.                        54
酒呑童子.シュテンドウジ.          53
亡者.モウジャ.                    46
病気.ビョウキ.                    34
烏天狗.カラステング.              30
猫又.ネコマタ.                    30
刀.カタナ.                        28
大蛇.ダイジャ.                    27
女官.ニョカン.                    26
疫神.エキシン.                    26
地震.ジシン.                      25
```

蛸.タコ.	25
烏帽子.エボシ.	23
簪.カンザシ.	23
頭巾.ズキン.	23
扇子.センス.	22
清姫.キヨヒメ.	20
化猫.バケネコ.	19
盃.サカズキ.	19
赤鬼.アカオニ.	19
一角.イッカク.	18
提灯.チョウチン.	18
雷神.ライジン.	18
ろくろ首.ロクロクビ.	17
子供.コドモ.	17
桃太郎.モモタロウ.	17
犬.イヌ.	17
虫.ムシ.	17
鯛.タイ.	17
麻疹.ハシカ.	17
兎.ウサギ.	16
土蜘蛛.ツチグモ.	16
火の玉.ヒノタマ.	16
炎.ホノオ.	16
蛇.ヘビ.	16
亀.カメ.	15
冠.カンムリ.	15
蝦蟇.ガマ.	15
髑髏.ドクロ.	15
化け猫.バケネコ.	14
老人.ロウジン.	14
茶釜.チャガマ.	14
閻魔.エンマ.	14
雉.キジ.	14
鬼女.キジョ.	14
眷属.ケンゾク.	13
首.クビ.	13
鹿島大明神.カシマダイミョウジン.	13
お歯黒.オハグロ.	12
入道.ニュウドウ.	12
猪.イノシシ.	12
老婆.ロウバ.	12
舞.マイ.	12
蟹.カニ.	12
雨.アメ.	12
妖怪.バケモノ.	11
妖怪.ヨウカイ.	11
文福茶釜.ブンブクチャガマ.	11
生首.ナマクビ.	11
蒲生川.ガモウガワ.	11
蝙蝠.コウモリ.	11
達磨.ダルマ.	11
鳥.トリ.	11
龍王.リュウオウ.	11
亡魂.ボウコン.	10
唐櫃.カラビツ.	10
嘴.クチバシ.	10
松明.タイマツ.	10
雷.カミナリ.	10
骸骨.ガイコツ.	10
鰐.ワニ.	10



このように、非常に多くの妖怪画像の説明に使われている主題キーワードと、ほんのわずかの妖怪画像の説明に使われている主題キーワードが存在していることがわかる。

パワーロー（べき乗則） [数学 I]

1 つの妖怪画に使われている主題キーワードの種類数を $f(1)$、

2 つの妖怪画に使われている主題キーワードの種類数を $f(2)$、

…

u 個の妖怪画に使われている主題キーワードの種類数を $f(u)$

としよう。

ここで、

$$f(u) = au^{-k}k$$

が成り立つ場合、これを冪乗則（べキ乗則、power law）という。この式の両辺について対数をとると、

$$\log f(u) \ = \log a - k \log u$$

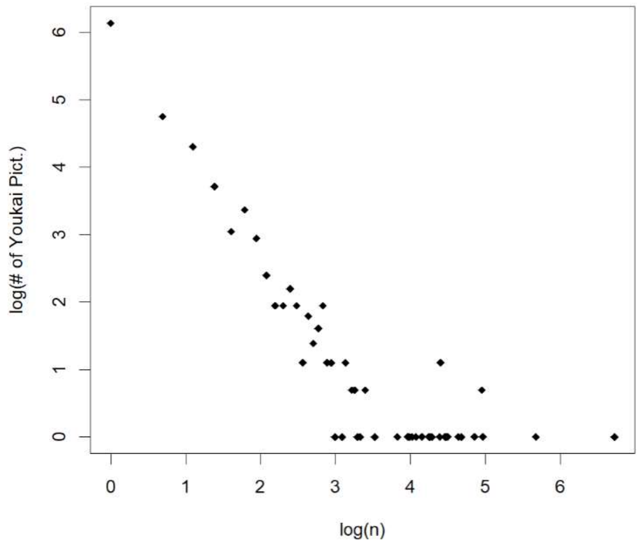
となるので、u 個の妖怪画に使われている主題キーワードの種類数 $f(u)$をともに対数をとれば線形の関係になる。このことをプログラミングにより実験してみよう。

ノック 26 本目

u 個の妖怪画に使われている主題キーワードの種類数 $f(u)$をともに対数をとれば線形の関係になればパワー則ということになる。この法則が妖怪画データの場合に成り立つか検討しよう。

RK26.R

```
dataYokai<-read.csv("DataYokai.csv",header=TRUE,row.names=1, fileEncoding = "shift-jis")
Yokailabel<-apply(dataYokai,2,sum)
sYokai<-sort(Yokailabel,decreasing=TRUE)
ndata<-table(sYokai)
nn<-as.integer(names(ndata))

logn<-log(nn)
logfn<-as.numeric(log(ndata))                #object def. from table type to numeric type

powerdata<-data.frame(logfn,logn)
summary(lm(logfn~logn,data=powerdata))

graphics.off()
plot(powerdata$logn,powerdata$logfn,pch=18,
        xlab="log(n)",ylab="log(# of Youkai Pict.)")
```




```
> summary(lm(logfn~logn,data=powerdata))
Call:
lm(formula = logfn ~ logn, data = powerdata)
Residuals:
    Min      1Q  Median      3Q     Max
-1.4421 -0.4392 -0.0797  0.3456  2.1926

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.3600     0.2975   14.66  < 2e-16 ***
logn        -0.9740     0.0828  -11.76 1.32e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7505 on 47 degrees of freedom
Multiple R-squared:  0.7465,    Adjusted R-squared:  0.7411
F-statistic: 138.4 on 1 and 47 DF,  p-value: 1.316e-15
```

ベキ分布の特徴は、中央値・最頻値が分布の左端に位置します。平均や分散という概念が事実上意味をなさないという点で、正規分布とは異なります。また、ベキ分布はどの尺度で拡大・縮小しても、常に同じような分布になるという「スケールの不変性」があります。となると、妖怪画を説明するための文字表現としての主題キーワードの使い方にはベキ乗則でできあがっているのかもしれません。また、妖怪画の多様性には、限界がないことを示しているのかもしれません。

ノック 27 本目

主題キーワードに「天狗」が入っている妖怪画の資源識別子をリストアップして、「天狗」を含む用語を列挙しよう。

RK27.R

```
dataYokai<-read.csv("DataYokai.csv",header=TRUE,row.names=1, fileEncoding = "shift-jis")
Yokailabel<-colnames(dataYokai)
tc<-regexpr("天狗",Yokailabel)      #天狗　鬼　女　男
targetcol<-Yokailabel[tc>0]
targetcol
```

　dataYokai には、行が各々の妖怪画と対応し、列が主題キーワードとなっている。例えば、資源識別子「A5_hermitage_0008_0001_0000」の妖怪画については、不動明王の列が 1 となっているため、この妖怪画はこの主題キーワードで説明がされていることを示している。

	不動明王.フドウミョウ.	不動明王.フドウミョウオウ.	乙姫.オトヒメ.	九尾.キュウビ.	九尾狐.キュウビノキツネ.	九頭竜.クズリュウ.	九頭竜権現.クズリュウゴンゲン.	九頭龍.クズリュウ.	亀.カメ.	事触れ.コトブレ.	五徳.ゴトク.	五輪塔.ゴリントウ.	五重塔.ゴジュウノトウ.
A5_hermitage_0001_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0001_0002_0000	0	0	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0001_0003_0000	0	0	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0003_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0005_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0006_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0007_0001_0000	0	0	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0007_0002_0000	0	0	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0007_0003_0000	0	0	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0008_0001_0000	0	1	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0008_0002_0000	0	0	0	0	0	0	0	0	0	0	0	0	
A5_hermitage_0008_0003_0000	0	0	0	0	0	0	0	0	0	0	0	0	

Yokailabel には、列名が格納されている。

```
> Yokailabel
  [1] "うさい角.ウサイカク."
  [2] "うわん.ウワン."
  [3] "おさん狐.オサンキツネ."
…
[870] "龍女.リュウニョ."
[871] "龍王.リュウオウ."
[872] "龍神.リュウジン."
```

regexpr("天狗", Yokailabel)により、Yokailabel に格納されている、各々の文字列について「天狗」が含まれている位置が得られます。また「天狗」が入っていない文字列では-1 となります。これが tc に格納されます。

そこで、Yokailabel[tc>0]とすれば、「天狗」が含まれている列名を抜き出すことができます。その結果を targetcol に格納します。

```
> targetcol
[1] "大天狗.ダイテング."        "天狗.テング."
[3] "烏天狗.カラステング."    "青天狗.アオテング."
[5] "鞍馬天狗.クラマテング."
```

ノック 28 本目

主題キーワードに「蛸」が入っている妖怪画の資源識別子をリストアップして、Google 検索してみよう。まず「蛸」が含まれる列名をピックアップし、その列名ごとに 1 となっている妖怪画の資源識別子を列挙してみましょう。

RK28.R

```
dataYokai<-read.csv("DataYokai.csv",header=TRUE,row.names=1, fileEncoding = "shift-jis")

Yokailabel<-colnames(dataYokai)
tc<-regexpr("蛸",Yokailabel)      #天狗　鬼　女　男
targetcol<-Yokailabel[tc>0]

targetmat<-dataYokai[,c(targetcol)]
sumtc<-apply(targetmat,1,sum)
targetmat<-targetmat[sumtc!=0,]

nf<-dim(targetmat)[2]

uu=list()
for(i in 1:nf){
  TFget<-targetmat[,i]!=0
  rnames<-rownames(targetmat[TFget,])
  uu[[i]]<-paste(c(colnames(targetmat)[i],rnames))
  print(uu[[i]])
}
```

出力結果

```
[1] "大蛸.オオダコ."
[2] "U426_nichibunken_0140_0001_0000"
 [1] "蛸.タコ."
 [2] "A5_pushkin_0018_0006_0000"
 [3] "A5_pushkin_0041_0001_0000"
 [4] "A5_pushkin_0041_0024_0000"
 [5] "U426_nichibunken_0053_0021_0000"
 [6] "U426_nichibunken_0053_0021_0004"
 [7] "U426_nichibunken_0073_0025_0000"
 [8] "U426_nichibunken_0092_0003_0003"
 [9] "U426_nichibunken_0094_0006_0000"
[10] "U426_nichibunken_0094_0006_0001"
[11] "U426_nichibunken_0097_0010_0000"
[12] "U426_nichibunken_0097_0021_0000"
[13] "U426_nichibunken_0104_0006_0000"
[14] "U426_nichibunken_0107_0008_0006"
[15] "U426_nichibunken_0108_0001_0005"
[16] "U426_nichibunken_0108_0002_0004"
[17] "U426_nichibunken_0123_0003_0002"
[18] "U426_nichibunken_0140_0001_0000"
[19] "U426_nichibunken_0142_0010_0000"
[20] "U426_nichibunken_0227_0009_0000"
```

```
[21] "U426_nichibunken_0228_0010_0000"
[22] "U426_nichibunken_0230_0001_0000"
[23] "U426_nichibunken_0230_0010_0000"
[24] "U426_nichibunken_0254_0008_0000"
[25] "U426_nichibunken_0321_0003_0000"
[26] "U426_nichibunken_0423_0001_0033"
```

このようにして得られた資源識別子をググると「蛸」の妖怪に到達できます。やってみよう!

ノック 29 本目

ノック 25 本目の結果を参考に、主題キーワードの好きな文字列により妖怪画の資源識別子をリストアップして、Google 検索してみよう。

ノック 30 本目

主題キーワードに「天狗」を含んでいる妖怪画について、主題キーワードの包含関係による使用の類似性を検討してみよう。

R30.R

```
1  dataYokai<-read.csv("DataYokai.csv",header=TRUE,row.names=1, fileEncoding = "shift-jis")
2  Yokailabel<-colnames(dataYokai)
3  tc<-regexpr("天狗",Yokailabel)      #天狗　鬼　女　男
4  targetcol<-Yokailabel[tc>0]
5  targetmat<-dataYokai[,c(targetcol)]
6  dim(targetmat)
7  sumtc<-apply(targetmat,1,sum)
8  targetmat<-targetmat[sumtc!=0,]
9  
10 if(!require(proxy)){
11   install.packages("proxy");library(proxy)
12 }
13 DataS<-t(targetmat)
14 nn<-apply(DataS,1,sum)
15 newL<-paste0(rownames(DataS),nn)
16 rownames(DataS)<-newL
17 
18 distName<-"Simpson"
19 hclustName<-"ward.D2"
20 DD<-dist(DataS,method=distName) #Jaccard Dice
21 
22 plot(hclust(DD,method=hclustName),
23      main=paste("dist type=",distName,"clust type=",hclustName))
```

1-9 行目は、RK29.R と同じです。targetmat は、行には妖怪画の資源識別子、列には「天狗」が入った主題キーワードからなる行列となっています。

	大天狗. ダイテング.	天狗. テング.	烏天狗. カラステング.	青天狗. アオテング.	鞍馬天狗. クラマテング.
A5_hermitage_0007_0001_0000	0	0	1	0	0
A5_pushkin_0005_0001_0000	0	1	0	0	0
…					
…					
U426_nichibunken_0442_0036_0000	0	1	0	0	0

ここで、天狗に関わる妖怪画は、88 作品ありました。
これを転置（行と列をひっくりかえす）すると、下の表になります。

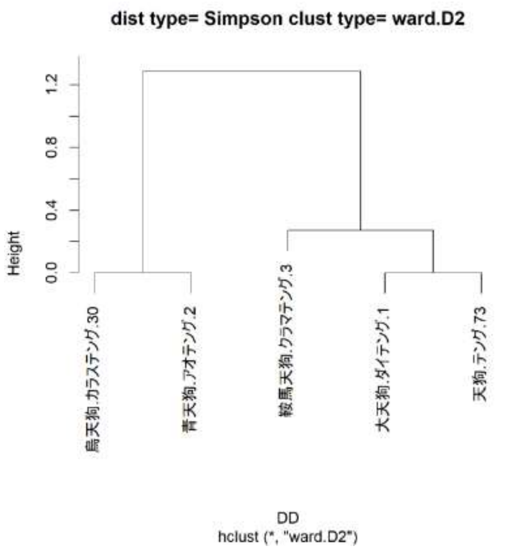
	A5_hermitage_0007_0001_0000	A5_pushkin_0005_0001_0000	…	…	U426_nichibunken_0442_0036_0000
大天狗. ダイテング.	0	0	…	…	0
天狗. テング.	0	1	…	…	1
烏天狗. カラステング.	1	0	…	…	0
青天狗. アオテング.	0	0	…	…	0
鞍馬天狗. クラマテング.	0	0	…	…	0

主題キーワード間の類似性として、ここでは、Simpson 係数（Overlap 係数）で表しました。
二つの主題キーワード A, B について、A と B を使われている妖怪画の総数をそれぞれN_AとN_Bとします。また A、B それぞれの両方のキーワードが用いられている妖怪画の総数を$N_{A\cap B}$とします。

$$overlap係数(A,B) = \frac{N_{A\cap B}}{\min{(N_A, N_B)}}$$

二つの主題キーワード A, B の使用頻度（N_AとN_B）の影響を受けにくいからですが、他の距離係数を用いることも可能です。

怪異・妖怪画像データベースには、天狗を含む主題キーワードとして 5 種類あり、青天狗は烏天狗に含まれ、一方、鞍馬天狗、大天狗は天狗に含まれるという傾向がみられます。つまり、妖怪画を説明するための主題キーワードの使い方には大きく二つのグループに従って説明されていることがわかります。



（烏天狗と青天狗）（鞍馬天狗、大天狗、天狗）はそれぞれ共通に妖怪画の説明に用いられていることがわかります。

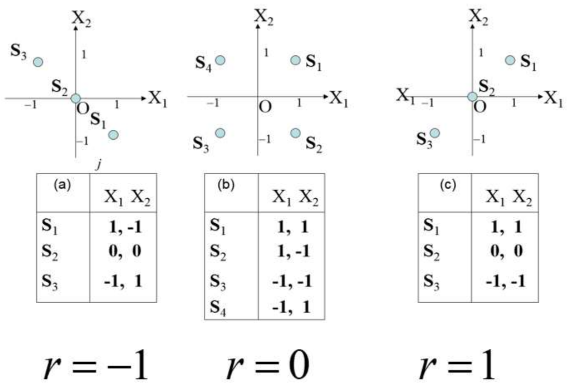
4. 関係性とは [数学 I]

「関係がある」というのを表す指標に、ピアソン相関係数があります。一般には相関係数と呼ばれています。以下の行列について二つの列 j と j' (変数 j と j')の相関係数を求めます。相関係数 $r_{jj'}$の式をみると、分母は、j と j'番目の列それぞれについて、平均値との差の 2 乗の和をルートを取った形になっています。一方、分子は、列ごとに j 番目と j'番目の平均値との差をとり掛け合わせた合計となっています。二つの変数間の関係をプロットした場合に、このピアソン相関では、右下がりの直線上にプロットされれば-1、右上がりにプロットされれば 1 となります。また、r=0 のときには二つの変数間に関係がないことがわかります。ピアソン相関係数を活用してデータ間の相関をみてみましょう。

$$\begin{pmatrix} x_{11} & x_{11} & \cdots & x_{1j} & \cdots & x_{1j'} & \cdots & \cdots & x_{1M} \\ x_{21} & x_{21} & \cdots & x_{2j} & \cdots & x_{2j'} & \cdots & \cdots & x_{2M} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & \cdots & x_{ij} & \cdots & x_{ij'} & \cdots & \cdots & x_{iM} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{N1} & x_{N1} & \cdots & x_{Nj} & \cdots & x_{Nj'} & \cdots & \cdots & x_{NM} \end{pmatrix}$$

ピアソンの相関係数

$$r_{jj'} = \frac{\sum_{i=1}^{N}(x_{ij}-\bar{x}_j)(x_{ij'}-\bar{x}_{j'})}{\sqrt{\sum_{i=1}^{N}(x_{ij}-\bar{x}_j)^2 \sum_{i=1}^{N}(x_{i'j}-\bar{x}_{j'})^2}} \qquad \bar{x}_j = \frac{\sum_{i=1}^{N} x_{ij}}{N}$$



ノック 31 本目

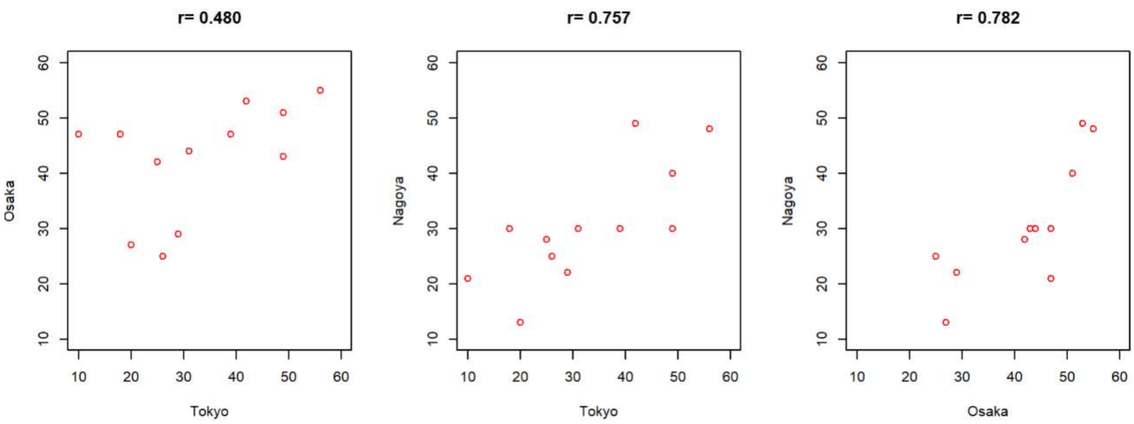
2007 年から 2018 年(Year)のそれぞれの年に、最低気温が 25℃以上（熱帯夜）であった日数を Tokyo、Osaka、Nagoya で集計した。Tokyo と Osaka, Tokyo と Nagoya、Osaka と Nagoya の間に相関があるだろうか。

Year	7	8	9	10	11	12	13	14	15	16	17	18
Tokyo	31	25	20	56	49	49	39	29	26	10	18	42
Osaka	44	42	27	55	51	43	47	29	25	47	47	53
Nagoya	30	28	13	48	40	30	30	22	25	21	30	49

RK31.R

```
Year<-   c( 7, 8, 9,10,11,12,13,14,15,16,17,18)
Tokyo<- c(31,25,20,56,49,49,39,29,26,10,18,42)
Osaka<- c(44,42,27,55,51,43,47,29,25,47,47,53)
Nagoya<-c(30,28,13,48,40,30,30,22,25,21,30,49)

dataSet<-data.frame(Year,Tokyo,Osaka,Nagoya)

TO<-paste("r=",substring(cor(dataSet[,2],dataSet[,3]),1,5))
```

```
TN<-paste("r=",substring(cor(dataSet[,2],dataSet[,4]),1,5))
ON<-paste("r=",substring(cor(dataSet[,3],dataSet[,4]),1,5))

par(mfrow=c(1,3))
plot(dataSet[,2],dataSet[,3],type='p',
      col=c("red"),xlab="Tokyo",ylab="Osaka",
      xlim=c(10,60),ylim=c(10,60),
      main=TO)

plot(dataSet[,2],dataSet[,4],type='p',
      col=c("red"),xlab="Tokyo",ylab="Nagoya",
      xlim=c(10,60),ylim=c(10,60),
      main=TN)

plot(dataSet[,3],dataSet[,4],type='p',
      col=c("red"),xlab="Osaka",ylab="Nagoya",
      xlim=c(10,60),ylim=c(10,60),
      main=ON)

cor(dataSet)
```




```
> cor(dataSet)
              Year      Tokyo      Osaka    Nagoya
Year    1.00000000 -0.2061740 0.08951652 0.1350897
Tokyo  -0.20617398  1.0000000 0.48094702 0.7571719
Osaka   0.08951652  0.4809470 1.00000000 0.7828882
Nagoya  0.13508968  0.7571719 0.78288818 1.0000000
```

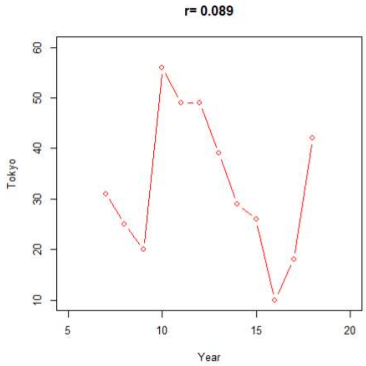
ノック 32 本目

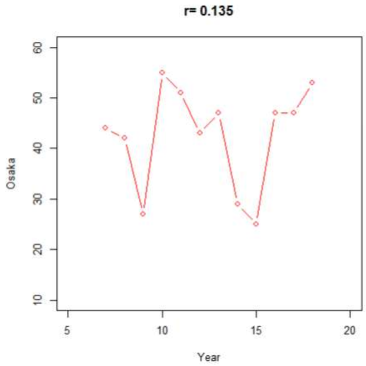
2007 年から 2018 年(Year)のそれぞれの年に、最低気温が 25℃以上（熱帯夜）であった日数を Tokyo、Osaka、Nagoya で集計した。年度と Tokyo、年度と Osaka，年度と Tokyo の間に相関があるだろうか。

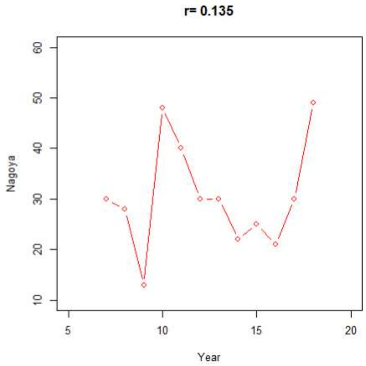
Year	7	8	9	10	11	12	13	14	15	16	17	18
Tokyo	31	25	20	56	49	49	39	29	26	10	18	42
Osaka	44	42	27	55	51	43	47	29	25	47	47	53
Nagoya	30	28	13	48	40	30	30	22	25	21	30	49

RK32.R

```
Year<-   c( 7, 8, 9,10,11,12,13,14,15,16,17,18)
Tokyo<- c(31,25,20,56,49,49,39,29,26,10,18,42)
Osaka<- c(44,42,27,55,51,43,47,29,25,47,47,53)
Nagoya<-c(30,28,13,48,40,30,30,22,25,21,30,49)

dataSet<-data.frame(Year,Tokyo,Osaka,Nagoya)


TO<-paste("r=",substring(cor(dataSet[,1],dataSet[,2]),1,5))
TN<-paste("r=",substring(cor(dataSet[,1],dataSet[,3]),1,5))
ON<-paste("r=",substring(cor(dataSet[,1],dataSet[,4]),1,5))

par(mfrow=c(1,3))
plot(dataSet[,1],dataSet[,2],type='b',
       col=c("red"),xlab="Year",ylab="Tokyo",
       xlim=c(5,20),ylim=c(10,60),
       main=TN)

plot(dataSet[,1],dataSet[,3],type='b',
       col=c("red"),xlab="Year",ylab="Osaka",
       xlim=c(5,20),ylim=c(10,60),
       main=ON)

plot(dataSet[,1],dataSet[,4],type='b',
       col=c("red"),xlab="Year",ylab="Nagoya",
       xlim=c(5,20),ylim=c(10,60),
       main=ON)

cor(dataSet)
```







以下の結果より、Year と 3 つの地域(Tokyo，Osaka，Nagoya)の相関係数は低いことがわかる。熱帯夜の日数が増えるか年とともに増えるかというとそうでもない。

```
> cor(dataSet)
              Year     Tokyo      Osaka    Nagoya
Year    1.00000000 -0.2061740 0.08951652 0.1350897
Tokyo  -0.20617398  1.0000000 0.48094702 0.7571719
Osaka   0.08951652  0.4809470 1.00000000 0.7828882
Nagoya  0.13508968  0.7571719 0.78288818 1.0000000
```

単回帰モデル：対数変換の面白さ [数学 I, [数学 B]

$$100000 = 10^5$$

について底を 10 として対数をとると

$$log_{10}(100000) = 5$$

となる。なんてことはない、100000 の桁数を表したことと変わりはない。

ところが、仮に

$$x^n x^m = 5$$

が成り立っているとしよう。この関係を見つけるのはグラフに書くとちょっと面倒だ。

ところが両辺、対数をとってみよう。

$$log_{10}(x^n y^m) = log5$$

これを変形すると、

$$nlog_{10}x + mlog_{10}y = log5$$

$log_{10}x$と$log_{10}y$の関係は線形になり、これらの関係の式をみつけるのが楽になる。

なお、R では、底を 2 とする場合は、log2( )、底を 10 とする場合は、log10( )、自然対数は log( )として計算できます。

ノック 33 本目

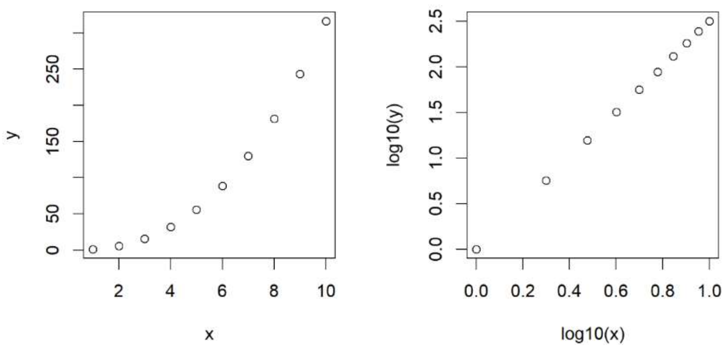
```
x<-c(1,2,3,4,5,6,7,8,9,10)
y<-c(1.00, 5.65, 15.58, 32.00, 55.90, 88.18, 129.64,181.01, 243.00, 316.22)
```

について $log_{10}x$、$log_{10}y$と各変数を対数変換して、x と y の関係を式で表してみよう。

RK33.R

```
1 x<-c(1,2,3,4,5,6,7,8,9,10)
2 y<-c(1.00, 5.65, 15.58, 32.00, 55.90, 88.18, 129.64,181.01, 243.00, 316.22)
3 par(mfrow=c(1,2))
4
5 plot(x,y)
6 plot(log10(x),log10(y))
7 datalog<-data.frame(logx=log10(x),logy=log10(y))
8 summary(lm(logy~logx,data=datalog))
```

6-7 行目：x と y の関係、$log_{10}x$、$log_{10}y$の関係を図に示しました。



この図をみると、$log_{10}x$と$log_{10}y$の間には線形の関係があることがわかります。

8 行目：

$log_{10}y$ = [切片] + [傾き]・$log_{10}x$

として、回帰モデルを作成した結果を以下に示す。この結果の見方を説明すると、赤字のところの(Intercept)が[切片] =-0.0005645、$log_{10}x$の[傾き]が logx の[傾き]=1.5005229 です。そこで、これら係数の統計的に有意性は Pr(>|t|)に示されていています。切片-0.0005645 には Pr(>|t|) =0.0976 であり、基準 p 値を 0.05 と設定卯すると、この p 値より大きいので、符号に意味はありません。つまり、0 とみなせます。一方、1.5005229 は Pr(>|t|)=2e-10 であり、p=0.05 より小さいので、この係数は有意に正であるとなります。式全体は、分散分析により p-value: < 2.2e-16 となっているので、式は成り立っていることになります(F-statistic の項目を見て、p-value が 0.05 より小さければ OK と確認することだけ理解しましょう。)

```
Call:
lm(formula = logy ~ logx, data = datalog)

Residuals:
       Min         1Q     Median         3Q        Max
-8.888e-04 -4.940e-05  3.454e-05  1.685e-04  5.645e-04

Coefficients:
              Estimate Std. Error   t value   Pr(>|t|)
(Intercept) -0.0005645  0.0003010    -1.875   0.0976 .
logx         1.5005229  0.0004168  3599.676   <2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0003981 on 8 degrees of freedom
Multiple R-squared:      1,    Adjusted R-squared:      1
F-statistic: 1.296e+07 on 1 and 8 DF,  p-value: < 2.2e-16
```

そこで、おおよそ、

$$log_{10}y = 0.000 + 1.500・log_{10}x$$

となります。これをもとに、

$$log_{10}y - 1.500 log_{10}x = 0$$

$$log_{10}(\frac{y}{x^{1.5}}) = 0$$

$$\frac{y}{x^{1.5}} = 10^0 = 1$$

結局

$$y = x^{1.5} = x^{3/2} = x\sqrt{x}$$

という関係が導かれました。

ノック 34 本目

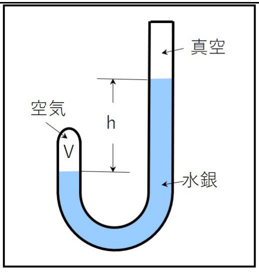
```
x<-c(1.000,0.933,0.895,0.870,0.851,0.835,0.823,0.812)
y<-c(1.000,2.143,3.348,4.594,5.873,7.177,8.503,9.849)
```

について $log_{10}x$、$log_{10}y$と各変数を対数変換して、x と y の関係を式で表してみよう。

RK33.R を参考にプログラムをつくって x と y の関係を式で表してみよう。

気体の体積と圧力の関係 [理数探究基礎]

かつて 17 世紀にイギリスの科学者ボイルは、気体の体積と圧力の関係を調べる実験を行った。J 字状の管の先端に閉じ込めた空気の体積 *V* と、管に入れた水銀の量の関係を調べた。温度は一定である。液面の高さの差 h から閉じ込められた空気にかかる圧力 *p* を求めた(表)。この表をもとに *p* と *V* の関係を求めてみよう。



体積 V(cm^3)	48.0	40.0	32.0	24.0	20.0	16.0	12.0
圧力 p(x10^{3}hPa)	1.01	1.23	1.54	2.04	2.46	3.04	4.09

法則性を見出すときには、対数プロットが役に立つことがある。底を 10 として二つの変数 *V* と *p* の対数をとり、直線関係が得られた。

$$log_{10}V = a + b\ log_{10}p$$

この式は、

$$log_{10}V +\ log_{10}p^{-b} = a$$

となり、さらに変形すると

$$log_{10}(Vp^{-b}) = a$$

$$Vp^{-b} = 10^a$$

と変形できるため、*a* と *b* が得られれば *V* と *p* の関係を式で表現できます。

(a)いま、x と y が比例関係にあるとき

$$y = x$$

対数をとると、傾き 1、切片(定数)0 の直線が得られます。

$$log_{10}y =\ log_{10}x$$

$$log_{10}\frac{y}{x} = 0$$

$$\frac{y}{x}=10^0 = 1$$

となる。

(b) $y = x^m$のとき

$$log_{10}y = \ mlog_{10}x$$

となるため、y と x の傾きが m、切片（定数）は 0 となります。よって、傾きから m を求められます。

ノック 35 本目

手計算で上記の枠内式を誘導して、y と x の関係を式であらわしてみましょう。（R プログラムはありません）

ノック 36 本目

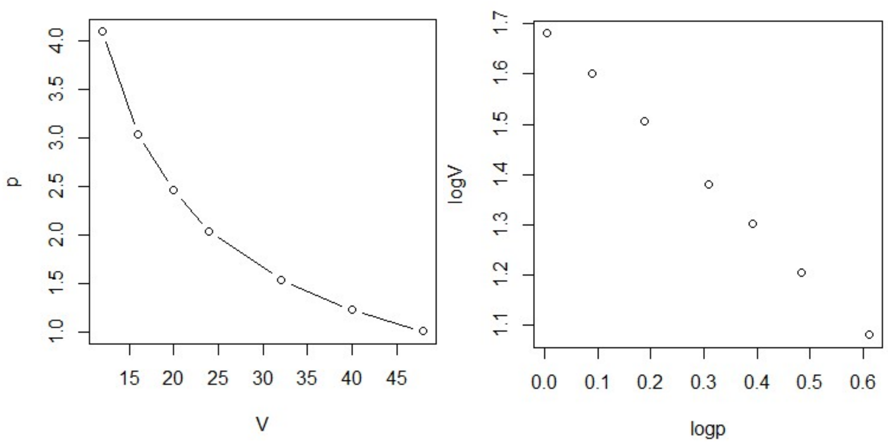
ノック 35 本目をもとに、

[1] V と p を、 x 軸と y 軸に設定しプロットしてみましょう。

[2] $log_{10}V$ と$log_{10}p$を、 x 軸と y 軸に設定しプロットしてみましょう。

RK36.R

```
V<-c(48.0,40.0,32.0,24.0,20.0,16.0,12.0)
p<-c(1.01,1.23,1.54,2.04,2.46,3.04,4.09)

par(mfrow=c(1,2))
#[1]
plot(V,p,type="b")
#[2]
logp<-log(p)
logV<-log(V)
plot(logp,logV)
```



左側の図が[1] V と p の関係、右側の図が [2] $log_{10}V$と$log_{10}p$ の関係となります。この図をみていただくと、$log_{10}V$と$log_{10}p$の間には、

$$log_{10}V = a - blog_{10}p$$

という関係が成り立つ。ここで、a と b をどのように求めるのでしょうか。ここでよく使われるのが最小二乗法です。

最小二乗法 [数学 B]

いま、y と x の測定値があるとします。ここで、y と x の間には、

$$y = a + bx \tag{1}$$

という関係があるとします。N 個の測定値 $y=\{y_1, y_2, \cdots, y_i, \cdots, y_N\}$と $x=\{ x_1, x_2, \cdots, x_i, \cdots, x_N\}$を(1)に代入する。

$$y_1 = a + bx_1 + e_1$$

$$\cdots$$

$$y_i = a + bx_i + e_i$$

$$\cdots$$

$$y_N = a + bx_N + e_N$$

ここで、e_iは残差である。測定値 y_i と x_i がぴったり(1)の式に合うことはほぼないので残差を入れてた式とする。では、この式を残差e_iにより式を変形してみましょう。

$$e_i = y_i - (a + bx_i)$$

ここで i= 1, 2, 3,…,N である。

そこで、全体の 2 乗誤差を定義する。

$$\mathrm{E(a, b)} = {e_1}^2 + {e_2}^2 + \cdots + {e_N}^2 = \sum_{i=1}^{N} {e_i}^2$$

この式の$E(a, b)$を最小とする a, b を求める。

この式に、$e_i=y_i-(a + b\ x_i)$を代入してみましょう。

$$\mathrm{E(a, b)} = \{y_1 - (\mathrm{a} + \mathrm{b}x_1)\}^2 + \cdots + \{y_i - (\mathrm{a} + \mathrm{b}x_i)\}^2 + \cdots + \{y_N - (\mathrm{a} + \mathrm{b}x_N)\}^2 = \sum_{i=1}^{N} \{y_i - (\mathrm{a} + \mathrm{b}x_i)\}^2$$

$\mathrm{E(a, b)}$を最小にする係数 a,b を求めてみましょう。最後の$\sum_{i=1}^{N} \{y_i - (\mathrm{a} + \mathrm{b}x_i)\}^2$の項を見てみましょう。

$$\mathrm{E(a, b)} = \sum_{i=1}^{N} \{y_i - (\mathrm{a} + \mathrm{b}x_i)\}^2 = \sum_{i=1}^{N} \{y_i - \mathrm{a} - \mathrm{b}x_i\}^2$$

$$= \sum_{i=1}^{N} \{{y_i}^2 + a^2 + (\mathrm{b}x_i)^2 - 2ay_i - 2bx_iy_i + 2abx_i\}$$

$$= \sum_{i=1}^{N} y^2 + \sum_{i=1}^{N} a^2 + b^2 \sum_{i=1}^{N} {x_i}^2 - 2a \sum_{i=1}^{N} y_i - 2b \sum_{i=1}^{N} x_iy_i + 2ab \sum_{i=1}^{N} x_i$$

ここで、$\sum_{i=1}^{N} y^2$、$\sum_{i=1}^{N} y_i$、$\sum_{i=1}^{N} {x_i}^2$、$\sum_{i=1}^{N} x_i$、$\sum_{i=1}^{N} x_iy_i$は実験値から計算できるので定数とみなせる。

$$Y_2 = \sum_{i=1}^{N} y^2、Y_1 = \sum_{i=1}^{N} y_i、X_2 = \sum_{i=1}^{N} {x_i}^2、X_1 = \sum_{i=1}^{N} x_i、Z = \sum_{i=1}^{N} x_iy_i$$

また、

$$\sum_{i=1}^{N} a^2 = a^2 + a^2 + \cdots + a^2 = Na^2$$

となる。

$$\mathrm{E(a,b)} = Y_2 + Na^2 + b^2X_2 - 2aY_1 - 2bZ_1 + 2abX_1$$

この式をみると、$\mathrm{E(a,b)}$は変数 a と b についての 2 次式となる。a^2とb^2の係数は N と$X_1 = \sum_{i=1}^{N} x_i$はともに正なので最小値が存在する。これを簡単に求めるには、
$\mathrm{E(a,b)}$を片方の変数 b を固定して、a で微分した式を作り 0 とおく。これを偏微分といい、$\frac{\partial \mathrm{E(a,b)}}{\partial a}$とする。

$$\frac{\partial \mathrm{E(a,b)}}{\partial a} = 2Na - 2Y_1 + 2X_1b = 0$$

同様に、変数 a を固定して、b で微分した式を作り 0 とおく。

$$\frac{\partial \mathrm{E(a,b)}}{\partial b} = 2X_2b - 2Z_1 + 2X_1a = 0$$

これら二つの式をもとに *a*, *b* を算出する。すると、

$$a = \frac{X_2Y_1 - X_1Z}{NX_2 - {X_1}^2} = \frac{\sum_{i=1}^{N} {x_i}^2 \sum_{i=1}^{N} y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} x_i y_i}{N\sum_{i=1}^{N} {x_i}^2 - \left(\sum_{i=1}^{N} x_i\right)^2}$$

$$b = \frac{X_1Y_1 - NZ_1}{NX_2 - {X_1}^2} = \frac{\sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i - N\sum_{i=1}^{N} x_i y_i}{N\sum_{i=1}^{N} {x_i}^2 - \left(\sum_{i=1}^{N} x_i\right)^2}$$

と求まります。
R スクリプトにより a,b の係数を求めてみましょう。

$$log_{10}V = a + b\ log_{10}p$$

のように、y = a + b x として a,b を求め、このモデルに従って、任意の x から y を予測するモデルのこと線形回帰モデル(linear regression model)といいます。R プログラムの関数 lm()に x と y とデータセット指定すると簡単に求めることができます。関数 lm()と式を誘導して求めた回帰係数（a と b）を比較してみよう。

ノック 37 本目

V<-c(48.0,40.0,32.0,24.0,20.0,16.0,12.0)
p<-c(1.01,1.23,1.54,2.04,2.46,3.04,4.09)
について x=$\log_{10}$V、y=$\log_{10}$ p と変換した後、

[1]lm()関数を用いて y=a+b x としたときの回帰係数 a, b を求めよう。

[2]最小二乗法により導いた式

$$a=\frac{X_2Y_1-X_1Z}{NX_2-{X_1}^2}=\frac{\sum_{i=1}^{N}{x_i}^2\sum_{i=1}^{N}y_i-\sum_{i=1}^{N}x_i\sum_{i=1}^{N}x_iy_i}{N\sum_{i=1}^{N}{x_i}^2-\left(\sum_{i=1}^{N}x_i\right)^2}$$

$$b=\frac{X_1Y_1-NZ_1}{NX_2-{X_1}^2}=\frac{\sum_{i=1}^{N}x_i\sum_{i=1}^{N}y_i-N\sum_{i=1}^{N}x_iy_i}{N\sum_{i=1}^{N}{x_i}^2-\left(\sum_{i=1}^{N}x_i\right)^2}$$

を用いて a, b を求めてみよう。

R37.R

```
1  V<-c(48.0,40.0,32.0,24.0,20.0,16.0,12.0)
2  p<-c(1.01,1.23,1.54,2.04,2.46,3.04,4.09)
3  logp<-log10(p)
4  logV<-log10(V)
5  
6  #[1]regression model
7  dataYX<-data.frame(logp,logV)
8  reg<-lm(logp~logV,data=dataYX)
9  summary(reg)
10 confint(reg,level=0.95)
11 a1<-reg$coefficients[1]
12 b1<-reg$coefficients[2]
13 a1
14 b1
15 10^a1
16 #[2] from derivation
17 N<-length(V)
18 a2<-(sum(logV^2)*sum(logp)-sum(logV)*sum(logp*logV))/(N*sum(logV^2)-
19 sum(logV)^2)
20 b2<-(N*sum(logV*logp)-sum(logV)*sum(logp))/(N*sum(logV^2)-sum(logV)^2)
21 a2
22 b2
23 10^a2
```

3-4 行目：logp には $\log_{10}$p、 logV には $\log_{10}$V の値が格納されています。データを data.frame 型にしないと lm()関数で回帰モデルを作ることができないので、とりあえず data.frame(logp, logV)として dataXY に代入する。dataXY には以下のように、7 サンプル 1-7 について logp と logV の値が格納される。

```
> dataYX
         logp      logV
1 0.004321374 1.681241
2 0.089905111 1.602060
3 0.187520721 1.505150
4 0.309630167 1.380211
```

```
5 0.390935107 1.301030
6 0.482873584 1.204120
7 0.611723308 1.079181
```

7-10 行目：

logp を y(目的変数)、logV を x(説明変数)として、回帰モデルを作るスクリプトは、

```
reg<-lm(logp~logV,data=dataYX)
```

となる。lm()関数の中を見てみましょう。「~」左側が目的変数、右側が説明変数となる。ここでは、$logp = a + b\ logV$ という回帰モデルをつくる。得られた回帰モデルは reg に格納される。

```
summary(reg)
```

により、回帰モデルを得ることができる。また

```
confint(reg,level=0.95)
```

により、係数 a, b の 95%信頼区間が得らえる。

結果を見てみましょう。residuals はそれぞれのサンプルの残差を表している。また、coefficients から、

$$logp = 1.692274 - 1.001642 logV$$

というモデルができたことがわかります。ここで、係数の右側にある***は係数の統計有意水準を表す。***とは p<0.001 の統計有意性を示している（とってもいいモデルができた)。

F-statistic とは回帰モデル全体における統計有意性を示す指標である。これも p-value: 1.003e-10 と非常に小さい p 値なので$logp$と$logV$には、線形回帰モデルとして関連づいていることがわかる。

```
> summary(reg)
Call:
lm(formula = logp~logV, data = dataYX)

Residuals:
         1          2          3          4          5          6          7
-0.0039506  0.0023218  0.0028683 -0.0001662  0.0018274 -0.0033033  0.0004025

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.692274   0.007832   216.1 4.03e-11 ***
logV        -1.001642   0.005564  -180.0 1.00e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.002955 on 5 degrees of freedom
Multiple R-squared:  0.9998, Adjusted R-squared:  0.9998
F-statistic: 3.241e+04 on 1 and 5 DF,  p-value: 1.003e-10
```

つづいて、confint(reg,level=0.95)により係数 a, b の区間推定を行ってみましょう。95%信頼区間において、

切片 a は、1.672143 < a <　1.7124062

係数 b は -1.015944 < b < -0.9873409

となった。注目すべきは、係数 b は-1 とみなせるという点です。

```
> confint(reg,level=0.95)
                2.5 %     97.5 %
(Intercept)  1.672143  1.7124062
```

logV   -1.015944 -0.9873409

なお[1]関数 lm()に x と y 求めた係数 a1（切片）、b1(傾き)と。 [2] a と b の値を求める式を誘導した場合の係数を a2（切片）、b2(傾き)とすると

```
> b1
     logV
-1.001642
> 10^a1
(Intercept)
   49.23505
> a2
[1] 1.692274
> b2
[1] -1.001642
> 10^a2
[1] 49.23505
```

確かに、a1=a2, b1=b2 となる。

それでは、

logp = 1.692274 – 1.001642 logV

の式から p と V の関係を導いてみましょう。

$$log_{10}(p) = 1.692274 - 1.001642\, log_{10}(V)$$

$$log_{10}(p) + \ log_{e10}(V^{1.001642}) = 1.692274$$

$$log_{10}(pV^{1.001642}) = 1.692274$$

$$pV^{1.001642} = 10^{1.692274} = 49.23505$$

いま、係数 1.001642 は 95%信頼区間で 1 とみなせるので、おおよそ、この実験では、

$$PV = 49.23505(= 一定)$$

という結果が得られた。多分、化学の授業で気体の状態方程式として勉強されることと思います。

ノック 38 本目

2007 年から 2018 年(Year)のそれぞれの年に、最低気温が 25℃以上（熱帯夜）であった日数を Tokyo、Osaka、Nagoya で集計した。以下の二つの回帰モデルを作成しよう。

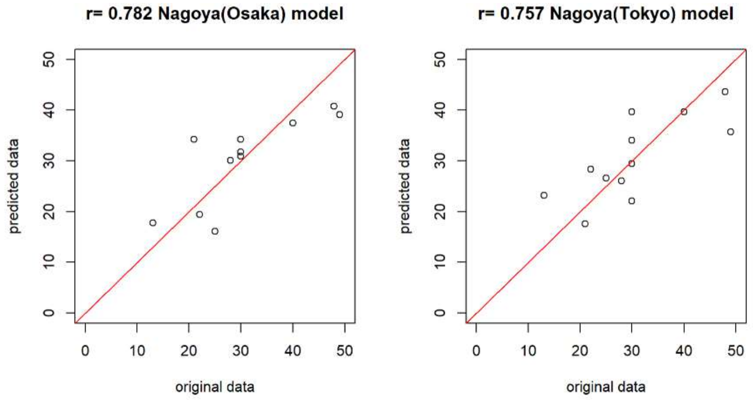
$$(熱帯夜の日数)_{\mathrm{Nagoya}} = a_0 + a_1(熱帯夜の日数)_{\mathrm{Osaka}}$$

$$(熱帯夜の日数)_{\mathrm{Nagoya}} = a_0 + a_1(熱帯夜の日数)_{\mathrm{Tokyo}}$$

Year	7	8	9	10	11	12	13	14	15	16	17	18
Tokyo	31	25	20	56	49	49	39	29	26	10	18	42
Osaka	44	42	27	55	51	43	47	29	25	47	47	53
Nagoya	30	28	13	48	40	30	30	22	25	21	30	49

RK38.R

```
Year<-   c( 7, 8, 9,10,11,12,13,14,15,16,17,18)
Tokyo<- c(31,25,20,56,49,49,39,29,26,10,18,42)
Osaka<- c(44,42,27,55,51,43,47,29,25,47,47,53)
Nagoya<-c(30,28,13,48,40,30,30,22,25,21,30,49)

dataSet<-data.frame(Year,Tokyo,Osaka,Nagoya)

summary(lm(Nagoya~Osaka,data=dataSet))
confint(lm(Nagoya~Osaka,data=dataSet))

summary(lm(Nagoya~Tokyo,data=dataSet))
confint(lm(Nagoya~Tokyo,data=dataSet))

lmobjNO<-lm(Nagoya~Osaka,data=dataSet)
lmobjNT<-lm(Nagoya~Tokyo,data=dataSet)

corNO<-substring(cor(dataSet$Nagoya,lmobjNO$fitted.value),1,5)
NO<-paste("r=",corNO,"Nagoya(Osaka) model")

corNT<-substring(cor(dataSet$Nagoya,lmobjNT$fitted.value),1,5)
NT<-paste("r=",corNT,"Nagoya(Tokyo) model")

par(mfrow=c(1,2))

plot(dataSet$Nagoya,lmobjNO$fitted.value,
     xlim=c(0,50),ylim=c(0,50),
     xlab="original data",ylab="predicted data",
     main=NO)
abline(a=0,b=1,col="red")

plot(dataSet$Nagoya,lmobjNT$fitted.value,
     xlim=c(0,50),ylim=c(0,50),
     xlab="original data",ylab="predicted data",
     main=NT)
abline(a=0,b=1,col="red")
```




```
> summary(lm(Nagoya~Osaka,data=dataSet))

Call:
lm(formula = Nagoya ~ Osaka, data = dataSet)

Residuals:
    Min      1Q  Median      3Q     Max
-13.196  -4.196  -1.321   3.750   9.875

Coefficients:
                Estimate   Std.   Error  t value Pr(>|t|)
(Intercept)     -4.4094     8.9989  -0.490   0.6347
Osaka            0.8214     0.2064   3.979   0.0026 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.942 on 10 degrees of freedom
Multiple R-squared:  0.6129,    Adjusted R-squared:  0.5742
F-statistic: 15.83 on 1 and 10 DF,  p-value: 0.002604

> confint(lm(Nagoya~Osaka,data=dataSet))
                  2.5 %    97.5 %
(Intercept) -24.4602481 15.641504
Osaka         0.3614589  1.281335
```

[1] (熱帯夜の日数)$_{\text{Nagoya}}$ $= a_0 + a_1$(熱帯夜の日数)$_{\text{Osaka}}$

summary(lm( ))によるa_0とa_1により

(熱帯夜の日数)$_{\text{Nagoya}}$ $= -\ 4.4094 + 0.8214$ x (熱帯夜の日数)$_{\text{Osaka}}$

という関係が得られました。

F-statistic: 15.83 on 1 and 10 DF,   p-value: 0.002604

により、F 統計量における p 値=0.002604 は、p<0.05 であるので、ここで得られた回帰モデ

ルは統計的に有意です。

続いて、それぞれの係数の t 統計量における p 値は、0.6347 および 0.0026 である。いま、p<0.05 のとき、係数が統計的に有意だとすると、a_0は統計的に有意でない、a_1は統計的に有意である。つまり、$a_0 = -4.44094$は 0 とみなせる。一方で$a_1 = -0.8214$は 0 とみなせない。ということを意味している。

confint(lm())の結果をみると、95%区間推定において

$-24.4602481 < a_0 < 15.641504$
$0.3614589 < a_1 < 1.281335$

であるので、a_0の区間は 0 をまたいでいるので、0 と見做されます。一方、a_1の区間は正と見做せます。同様に、$(\text{熱帯夜の日数})_{\text{Nagoya}} = a_0 + a_1(\text{熱帯夜の日数})_{\text{Tokyo}}$についても統計的に解釈してみてください。

```
> summary(lm(Nagoya~Tokyo,data=dataSet))

Call:
lm(formula = Nagoya ~ Tokyo, data = dataSet)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.2128  -4.5821   0.4305   3.6855  13.2948 

Coefficients:
             Estimate    Std.     Error t value Pr(>|t|)
(Intercept)  11.8560     5.5043    2.154  0.05668 .
Tokyo         0.5678     0.1549    3.666  0.00435 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.289 on 10 degrees of freedom
Multiple R-squared:  0.5733,    Adjusted R-squared:  0.5306 
F-statistic: 13.44 on 1 and 10 DF,  p-value: 0.004349

> confint(lm(Nagoya~Tokyo,data=dataSet))
                 2.5 %     97.5 %
(Intercept) -0.4082637 24.1203553
Tokyo        0.2226706  0.9130017
```

重回帰モデルを作成しよう

最小二乗法では、x と y のデータから

$y = a + b\,x$

の式を導いた。これを単回帰分析と言います。この単回帰分析をさらに発展させ、x_1、x_2 ,y のデータから

$y = a + b_1\,x_1 + b_2\,x_2$

という式を導くことができます。この場合、複数の説明変数(x_1、x_2)から目的変数(y)を求める重回帰モデルといいます。熱帯夜のデータを用いて重回帰モデルを構築してみましょう。

ノック 39 本目

複数の説明変数、$(\text{熱帯夜の日数})_{\text{Osaka}}$、$(\text{熱帯夜の日数})_{\text{Tokyo}}$から$(\text{熱帯夜の日数})_{\text{Nagoya}}$を説明する回帰モデルをつくってみよう。

$$(\text{熱帯夜の日数})_{\text{Nagoya}} = a_0 + a_1(\text{熱帯夜の日数})_{\text{Osaka}} + a_2(\text{熱帯夜の日数})_{\text{Tokyo}}$$

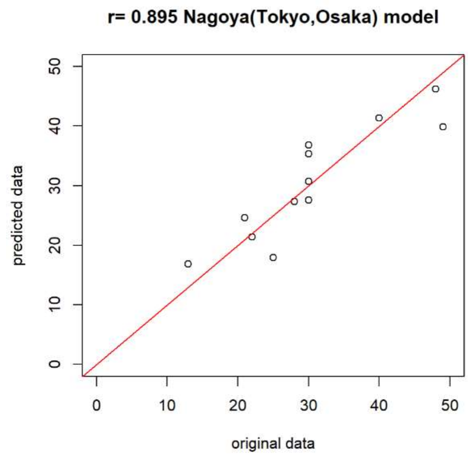
Year	7	8	9	10	11	12	13	14	15	16	17	18
Tokyo	31	25	20	56	49	49	39	29	26	10	18	42
Osaka	44	42	27	55	51	43	47	29	25	47	47	53
Nagoya	30	28	13	48	40	30	30	22	25	21	30	49

RK39.R

```
Year<-   c( 7, 8, 9,10,11,12,13,14,15,16,17,18)
Tokyo<- c(31,25,20,56,49,49,39,29,26,10,18,42)
Osaka<- c(44,42,27,55,51,43,47,29,25,47,47,53)
Nagoya<-c(30,28,13,48,40,30,30,22,25,21,30,49)

summary(lm(Nagoya~Tokyo+Osaka,data=dataSet))

lmobjNTO<-lm(Nagoya~Tokyo+Osaka,data=dataSet)
corNTO<-substring(cor(dataSet$Nagoya,lmobjNTO$fitted.value),1,5)
NTO<-paste("r=",corNTO,"Nagoya(Tokyo,Osaka) model")

graphics.off()
plot(dataSet$Nagoya,lmobjNTO$fitted.value,
      xlim=c(0,50),ylim=c(0,50),
      xlab="original data",ylab="predicted data",
      main=NTO)
abline(a=0,b=1,col="red")
```

>summary(lm(Nagoya~Tokyo+Osaka,data=dataSet))

```
Call:
lm(formula = Nagoya ~ Tokyo + Osaka, data = dataSet)
Residuals:
     Min       1Q   Median       3Q      Max
-6.7894 -3.6632 -0.0187  1.9237  9.0948

Coefficients:
            Estimate Std. Error t value   Pr(>|t|)
(Intercept)  -5.9828     6.8157  -0.878    0.4029
Tokyo         0.3714     0.1271   2.923    0.0170 *
Osaka         0.5715     0.1778   3.215    0.0106 *
---
Signif. codes:
  0 '***' 0.001 '**' 0.01 '*' 0.05   '.' 0.1 ' ' 1
Residual standard error: 5.241 on 9 degrees of freedom
Multiple R-squared:  0.8014,   Adjusted R-squared:  0.7573
F-statistic: 18.16 on 2 and 9 DF,  p-value: 0.0006932
```



$$(\text{熱帯夜の日数})_{\mathrm{Nagoya}} = -5.9829 + 0.3714\left(\text{熱帯夜の日数}\right)_{\mathrm{Osaka}} + 0.5715\,(\text{熱帯夜の日数})_{\mathrm{Tokyo}}$$

という回帰モデルが得られました。このように複数の変数[$\left(\text{熱帯夜の日数}\right)_{\mathrm{Osaka}}$ と $(\text{熱帯夜の日数})_{\mathrm{Tokyo}}$]により一つの変数[$(\text{熱帯夜の日数})_{\mathrm{Nagoya}}$] を説明する回帰モデルを、重回帰モデルと呼びます。

統計検定よりそれぞれの係数が 0 と見做せるかどうかを検定すると、p 値は 0.402, 0.017, 0.0106 です。つまり、95%信頼区間では -5.9829 は 0 と見做せます。一方、0.3714 と 0.5715 は 0 と見做せません。すなわち、$\left(\text{熱帯夜の日数}\right)_{\mathrm{Osaka}}$ と $(\text{熱帯夜の日数})_{\mathrm{Tokyo}}$は $(\text{熱帯夜の日数})_{\mathrm{Nagoya}}$ に正に寄与していることがわかります。

5. 確率・統計[数学 I, 数学 A]

計算機では、さまざまな乱数を発生させることができます。例えば、コインを投げて表と裏のでる確率を乱数を使って実験することができます。すると、100,000 回思考したときに、表と裏の出る確率が実際にどのくらいばらつくのか？についてもシミュレーション通して推定することができます。ここでは、乱数を使って、確率を理解しましょう。ちなみに高校では　数学 I ならびに数学 A の発展的内容です。

ノック 40 本目

おみくじシステムをつくろう。いま、4 種類のくじ、大吉(Daikichi)、吉(Kichi)、小吉(Shokichi)、凶(Kyo)、を考える。これをランダムに選ぶプログラムを作成しよう。

このことは、sample 関数を用いるとできる。以下のプログラムをみてみよう。2 行目で c("Daikichi","Kichi","Shokichi","Kyo")で、4 つの事象、大吉(Daikichi)、吉(Kichi)、小吉(Shokichi)、凶(Kyo)を定義し、その起こる確率を prob=c(1/4, 1/4, 1/4, 1/4)としているので、それぞれの起こる確率 P(Daikichi)=P(Kichi)=P(shokichi)=P(Kyo)=1/4 です。この確率で、ランダムに大吉(Daikichi)、吉(Kichi)、小吉(Shokichi)、凶(Kyo)を選び表示してくれます。ここで、replace=TRUE なので、くじを引いて後、元に戻す仕様です。

以下のプログラムを実行しよう。Daikichi が出るまで何度も実行してみよう。

RK40.R

```
1 n<-1
2 sample(c("Daikichi","Kichi","Shokichi","Kyo"),
3         prob = c(1/4,1/4,1/4,1/4),
4         size=n, replace = TRUE)
```

ノック 41 本目

1000 回おみくじを引いてみよう。発生する乱数が毎回異なるので、大吉(Daikichi)、吉(Kichi)、小吉(Shokichi)、凶(Kyo)の出現数が異なることを確かめよう。ただし、それぞれのおみくじの出現確率は 1/4 なので、おおよそ 250 回づつ出現することが確認できるでしょう。

RK41.R

```
graphics.off()
n<-1000
```

```
sData<-sample(c("Daikichi","Kichi","Shokichi","Kyo"),
              prob = c(1/4,1/4,1/4,1/4),
              size=n, replace = TRUE)
shukei<-table(sData)
shukei
barplot(shukei)
```

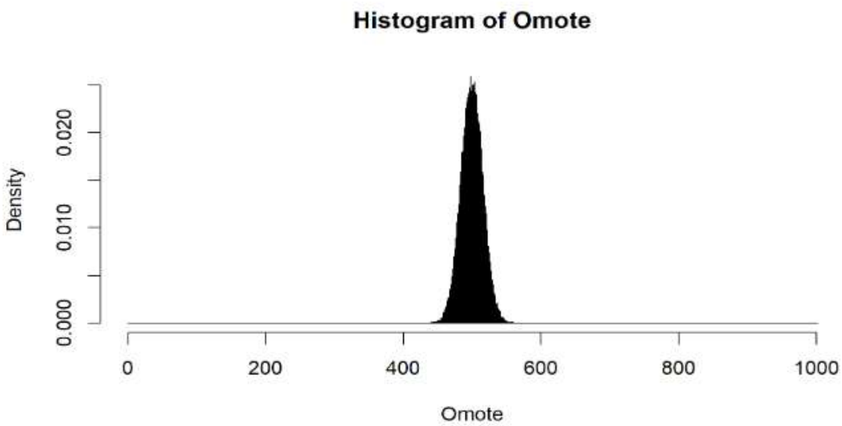
1000 回ランダムに選ぶと、Daikichi,、Kichi、Shokichi、Kyo はだいたい 250 回ずつになります。ただ乱数を使っているので毎回微妙にかわります。

ノック 42 本目

乱数を使って「コインを 1000 回投げる」これを 100000 回行ったとき、0-1000 回表が出る回数についてヒストグラムで表してみよう。

RK42.R

```
graphics.off()
n<-1000
m<-100000
Omote<-rep(0,m)

for(i in 1:m){
OU<-sample(c("Omote","Ura"),
              prob = c(1/2,1/2),
              size=n, replace = TRUE)
Omote[i]<-sum(OU=="Omote")
}
hist(Omote,seq(0,(n+1),1),freq=FALSE)
```



実は、これは二項分布で記述できます。

コインの表と裏の出る確率を p, q=1-p としましょう。N=1000 回コインを投げたと、k 回表が出る確率 $P(k)$とします。

$P(k) = {}_{N}C_{k}\ p^{k}q^{N-k}$
いま、p=q=1/2 なので

$$P(k) = {}_{N}C_{k}\ p^{N}$$

となります。

5 回、コイントスをしてみましょう(N=1)
一回も表が出ない確率は

$$P(0) = {}_{5}C_{0}\ \left(\frac{1}{2}\right)^{5} = \left(\frac{1}{2}\right)^{5}$$

1 回から 5 回表が出る確率は、

$$P(1) = {}_{5}C_{1}\ \left(\frac{1}{2}\right)^{5} = 5\left(\frac{1}{2}\right)^{5}$$
$$P(2) = {}_{5}C_{2}\ \left(\frac{1}{2}\right)^{5} = \frac{5}{2}\frac{4}{1}\left(\frac{1}{2}\right)^{5}$$
$$P(3) = {}_{5}C_{3}\ \left(\frac{1}{2}\right)^{5} = \frac{5}{2}\frac{4}{1}\left(\frac{1}{2}\right)^{5}$$
$$P(4) = {}_{5}C_{4}\ \left(\frac{1}{2}\right)^{5} = 5\left(\frac{1}{2}\right)^{5}$$
$$P(5) = {}_{5}C_{5}\ \left(\frac{1}{2}\right)^{5} = \left(\frac{1}{2}\right)^{5}$$

となります。ここで、${}_{N}C_{k}\ = \frac{N!}{k!(N-k)!}$ です。

$$P(0) + P(1) + \cdots + P(5) = (1 + 5 + 10 + 10 + 5 + 1)\left(\frac{1}{2}\right)^{5} = 32\left(\frac{1}{2}\right)^{5} = 1$$

と 0-5 回表が出る確率の合計も 1 となります。

これは、実は、$(a+b)^{n}$を展開すると、

$$(a+b)^{N} = {}_{N}C_{0}\ a^{N}b^{0} + {}_{N}C_{1}\ a^{N-1}b^{1} + \cdots + {}_{N}C_{k}\ a^{N-k}b^{k} + \cdots + {}_{N}C_{N-1}a^{1}b^{N-1} + {}_{N}C_{0}\ a^{0}b^{N}$$

ここで、$a = p$、$b = q = 1 - p$ とおくと、

$$1 = {}_{N}C_{0}\ p^{N}(1-p)^{0} + {}_{N}C_{1}\ p^{N-1}(1-p)^{1} + \cdots + {}_{N}C_{k}\ p^{N-k}(1-p)^{k} + \cdots + {}_{N}C_{N-1}p^{1}(1-p)^{N-1} + {}_{N}C_{N}\ p^{0}(1-p)^{N}$$

となり、第 1 項から N+1 項がそれぞれ、*P*(0)、*P*(1)、…、*P*(N)と対応します。

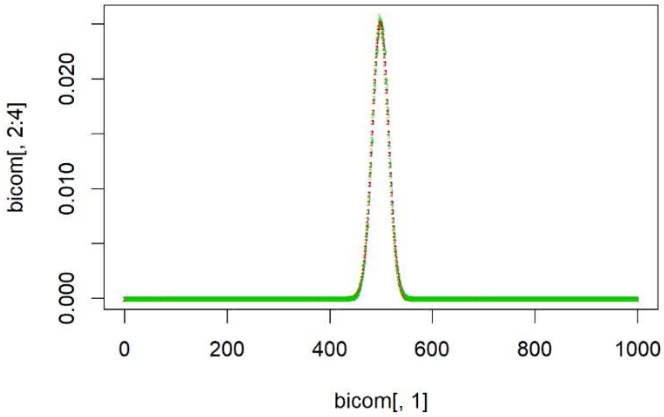
ノック 43 本目

乱数を使って「コインを 1000 回投げる」これを 100000 回行ったとき、0-1000 回表が出る回数についてヒストグラムについて二項分布、正規分布を比較しよう。 二項分布の平均値は、 $\mathrm{E(X)} = Np$ 分散は、 $\mathrm{V(X)} = Np(1-p)$ となります。これを使って正規分布の密度関数と比較できます。これを確認しましょう。

RK43.R

```
graphics.off()
n<-1000
m<-100000
Omote<-rep(0,m)

for(i in 1:m){
OU<-sample(c("Omote","Ura"),
              prob = c(1/2,1/2),
              size=n, replace = TRUE)
Omote[i]<-sum(OU=="Omote")
}
hist(Omote,seq(0,(n+1),1),freq=FALSE)

av<-n*1/2
sd<-sqrt(n*1/2*(1-1/2))

#-----------
rdens<-hist(Omote,seq(0,(n+1),1),freq=FALSE)$density
##
xaxis<-0:n

bi<-rep(0,n+1)
dn<-rep(0,n+1)
for(i in 0:n){
  bi[i]<-dbinom(i, size=n, prob=1/2, log = FALSE)
  dn[i]<-dnorm(i,mean=av,sd=sd)
}

bicom<-data.frame(xaxis,bi,dn,rdens)
matplot(bicom[,1],bicom[,2:4],cex=0.3)
```

プログラム中の bi 理論分布から導いた二項分布の密度分布(1 と表記)、dn は正規分布の密度分布(2 と表記)、の rdens はコインを 1000 回投げた時のコインの表と裏の相対数を 10000 回行った場合の分布(3 と表記)を示しています。



図をみると、黒（数字は 1）がランダムサンプリングによるコイントスで表が出る相対頻度、赤（数字の 2）と緑(数字の３)は、二項分布と正規分布の理論頻度です。結局のところ、試行回数を増やすと、ランダムサンプリングによるコイントスで表が出る相対頻度は、二項分布および正規分布とほぼ同一になります。このように試行数を 1000 回にするとほぼ同じ分布となります。こうやって、計算機実験ができる楽しさを実感しよう。

二項分布による統計検定 [数学 I]

ノック 44 本目

T 社と K 社でマスクを開発し、どちらを好むか調査した。20 人を無作為に抽出し、T 社と K 社のマスクしてもらい、どちらが好いか訊いた。T 社製が好いと言った人は 5 人、K 社製が好いと言った人は 15 人だった。このとき K 社と T 社のマスクの好みは同等か？

この課題を確率値（p 値）で評価する。まず仮説をたてる。

帰無仮説 H_0：T 社と K 社のマスクに着け心地などの差はない。

とする。これに対する対立仮説は 3 つ設定できる。

対立仮説 H_1:

(1) $x_T(=x) \neq x_K(=N-x)$

(2) $x_T(=x) < x_K(=N-x)$

(3) $x_T(=x) > x_K(=N-x)$

ところで、$x_T = 5$、$x_K = 15$であるので、

いま(3)は成り立たないので、(1)と(2)の検定を行う。

「T 社製が好い人は 5 人、K 社製が好い人は 15 人」ということは、見かけ上、K 社製が優勢である。これを統計検定する。まず、「K 社製が好い人が 15 人」ということを、「K 社製が好い人は少なくとも 15 人」と捉える。そこで、少なくとも 15 人だから、K 社製が好い人が 15-20 人の確率を求めることになる。ここで p=1/2 とする。

$$P(k) = {}_N C_k\ p^k (1-p)^{N-k}$$

について、N=20, p=0.5 を代入すると

$$P(k) = {}_N C_k\ 0.5^k (0.5)^{N-k} = {}_{20}C_k\ 0.5^{20}$$

となる。そこで、$P(15)$-$P(20)$の総和を求める。

$$P(15) + \cdots + P(20) = 0.5^{20} \{ {}_{20}C_{15} + {}_{20}C_{16} + {}_{20}C_{17} + {}_{20}C_{18} + {}_{20}C_{19} + {}_{20}C_{20} \} 0.5^{20}$$

これを R で計算してみよう。

RK44.R

```
Pr15<-choose(20,15)*(0.5^20)
Pr16<-choose(20,16)*(0.5^20)
Pr17<-choose(20,17)*(0.5^20)
Pr18<-choose(20,18)*(0.5^20)
Pr19<-choose(20,19)*(0.5^20)
Pr20<-choose(20,20)*(0.5^20)
Pr15+Pr16+Pr17+Pr18+Pr19+Pr20
```

```
> Pr15+Pr16+Pr17+Pr18+Pr19+Pr20
[1] 0.02069473
```

となり p <0.05 となり、H_0 は棄却され、H_1 (2) $x_T(=x) < x_K(=N-x)$が採択される。

一方、

$$P(15)+\cdots+P(20)=0.5^{20}\{{}_{20}C_{15}+{}_{20}C_{16}+{}_{20}C_{17}+{}_{20}C_{18}+{}_{20}C_{19}+{}_{20}C_{20}\}0.5^{20}$$

と

$$P(0)+\cdots+P(5)=0.5^{20}\{{}_{20}C_{0}\ +{}_{20}C_{1}\ +{}_{20}C_{2}\ +{}_{20}C_{3}\ +{}_{20}C_{4}\ +{}_{20}C_{5}\ \}0.5^{20}$$

の和を求めると、

0.02069473・2= 0.04138946

となり、これも p<0.05 であるので、H_0 は棄却され、H_1 (1) $x_T(=x) \neq x_K(=N-x)$が採択される。

この検定を二項検定といい、binom.test()により検定できる。

帰無仮説 H_0：T 社と K 社のマスクに着け心地などの差はない。

とする。これに対する対立仮説は３つ設定できる。

対立仮説 H_1:

(1) $x_T(=x) \neq x_K(=N-x)$

(2) $x_T(=x) < x_K(=N-x)$

(3) $x_T(=x) > x_K(=N-x)$

ところで、$x_T=5$、$x_K=15$である。binom.test()の関数は以下のように定義されている。

`binom.test(x,`	成功数
`n,`	全数
`p = 0.5,`	成功の仮定された確率
`alternative = c("two.sided", "less", "greater"),`	対立仮説:"two.sided", "greater", "less"
`conf.level = 0.95`	区間推定の有意水準（デフォルトは 0.95）
`)`	

3 種の対立仮説"two.sided", "greater", "less"で定義できる。以下のプログラムの[A]、[B]、[C]は上の対立仮説のどれと対応するか？プログラムを実行して検討しよう。

ノック 45 本目

T 社と K 社でマスクを開発し、どちらを好むか調査した。20 人を無作為に抽出し、T 社と K 社のマスクしてもらい、どちらが好いか訊いた。T 社製が好いと言った人は 5 人、K 社製が好いと言った人は 15 人だった。このとき K 社と T 社のマスクの好みは同等か？ これを 3 種の対立仮説"two.sided", "greater", "less"で定義して二項検定しよう。

RK45.R

```
binom.test(15,20,p=0.5,alternative="two.sided")   #[A]
binom.test(15,20,,p=0.5,alternative="greater")   #[B]
binom.test(15,20,,p=0.5,alternative="less")      #[C]
```

```
> binom.test(15,20,p=0.5,alternative="two.sided")

        Exact binomial test

data:  15 and 20
number of successes = 15, number of trials = 20, p-value = 0.04139
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.5089541 0.9134285
sample estimates:
probability of success
                  0.75

> binom.test(15,20,,p=0.5,alternative="greater")
        Exact binomial test
data:  15 and 20
number of successes = 15, number of trials = 20, p-value = 0.02069
alternative hypothesis: true probability of success is greater than 0.5
95 percent confidence interval:
 0.5444176 1.0000000
sample estimates:
probability of success
                  0.75
> binom.test(15,20,,p=0.5,alternative="less")
```

```
        Exact binomial test
data:   15 and 20
number of successes = 15, number of trials = 20, p-value = 0.9941
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
 0.0000000 0.8959192
sample estimates:
probability of success
                  0.75
```

6. 集合 [数学 I]

集合といえば、以下の様な感じです。

1 から 1000 までの整数について、2 の倍数、3 の倍数、 2 かつ 3 の倍数、 2 あるいは 3 のいずれの倍数でもない数を求めてみよう。

では、計算機では集合の問題をどうやって扱うのでしょうか？課題を解決しながら理解しましょう。

ノック 46 本目

1 から 1000 までの整数について、2 の倍数の数を求めよう。

RK46.R

```
1 N<-c(1:1000)
2 TF2<-(N%%2==0)
3 fold2<- N[TF2]
4 Nfold2<-length(fold2)
5 Nfold2
```

1 行目：N には 1 から 1000 の数字が格納されている。N[1]=1, N[2]=2, …, N[1000]=1000 である。
c(1:1000)は 1 から 1000 からなるベクトルを定義して N に格納する。

2 行目：N%%2==0 は、N を 2 で割った余り(%%)が 0 のとき TRUE(真)、0 でないときに FALSE(偽)となる。
TF2 には、FALSE, TRUE, FALSE, TRUE, …, FALSE, TRUE が格納される。
「A==B」は A と B が同じとき TRUE となる。一方「A!=B」とすると、A と B が異なるとき TRUE となる。

3 行目：N[TF2]により、TF2 が TRUE の要素が fold2 に格納される。fold2[1]=2, fold2[2]=4, …, fold2[500]=1000 となる。

4 行目：length()により要素の数を求めることできる。要素数 Nfold2 は 500 となる。

このようにして 1 から 1000 までの整数では、2 で割り切れる整数は 500 個あることがわかった。

ノック 47 本目

1 から 20 までの整数について、2 の倍数からなる集合を A、3 の倍数からなる集合を B とする。二つの集合の要素数によりベン図を作成しよう。

RK47.R

```
#ben diagram
N<-c(1:20)
N[N%%2==0 & N%%3!=0]
N[N%%2!=0 & N%%3==0]
N[N%%2==0 & N%%3==0]
N[N%%2!=0 & N%%3!=0]
length(N[N%%2==0 & N%%3!=0])
length(N[N%%2!=0 & N%%3==0])
```

```
 9  length(N[N%%2==0 & N%%3==0])
10  length(N[N%%2!=0 & N%%3!=0])
```

1 行目：#ben diagram はコメント行です。#をつけると行の中でそれ以降は実行されません。

2 行目：N には 1,2,3,…,20 が格納されます。

3 行目：length(N[N%%2==0 & N%%3!=0])について、

　N%%2==0 はベクトル N の要素について 2 で割った余りが 0 であれば TRUE、

　N%%3!=0 はベクトル N の要素について 3 で割った余りが 0 でなければ TURE となります。

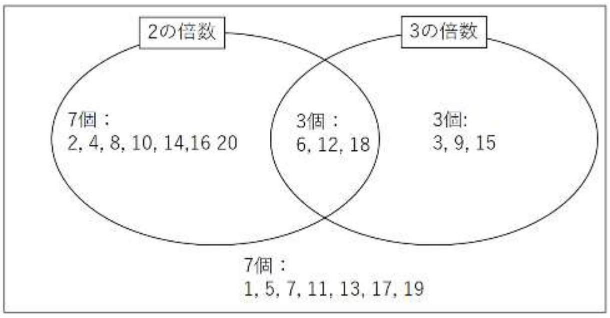
これらの二つの条件が「&」であるので、「N%%2==0」かつ「N%%3!=0」が成り立てば TRUE となります。以下の表では、T は TRUE、F は FALSE を表しています。空欄をうめてみよう。

要素	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
N%%2==0	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T
N%%3!=0	T	T	F	T	T	F	T	T	F	T	T	F	T	T	F	T	T	F	T	T
N%%2==0& N%%3!=0	F	T	F	T	F	F	F	T	F	T	F	F	F	T	F	T	F	F	F	T
N%%2!=0 & N%%3==0																				
N%%2==0& N%%3==0																				
N%%2!=0 & N%%3!=0																				

となり、N[N%%2==0 & N%%3!=0]は{2,4,8,10,14,16,20}の 7 個の要素からなるベクトルとなります。同様に、N[N%%2!=0 & N%%3==0]、N[N%%2!=0 & N%%3==0]、N[N%%2==0 & N%%3==0]、N[N%%2!=0 & N%%3!=0]の要素を求めよう。

　出力結果をみると

```
> N<-c(1:20)
> N[N%%2==0 & N%%3!=0]
[1]  2  4  8 10 14 16 20
> N[N%%2!=0 & N%%3==0]
[1]  3  9 15
> N[N%%2==0 & N%%3==0]
[1]  6 12 18
> N[N%%2!=0 & N%%3!=0]
[1]  1  5  7 11 13 17 19
> length(N[N%%2==0 & N%%3!=0])
[1] 7
> length(N[N%%2!=0 & N%%3==0])
[1] 3
> length(N[N%%2==0 & N%%3==0])
[1] 3
> length(N[N%%2!=0 & N%%3!=0])
[1] 7
```



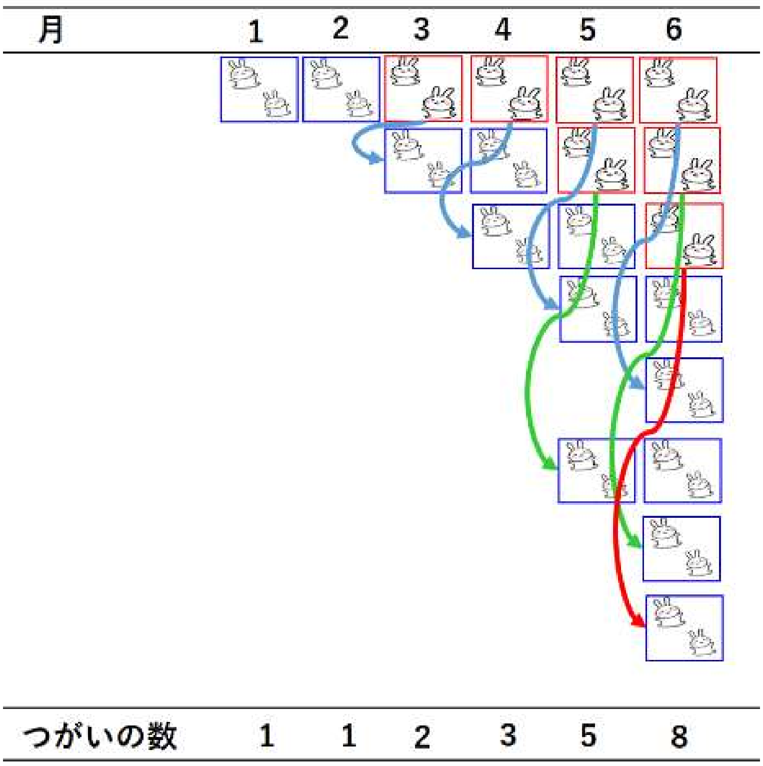
となる。というわけで、ベン図に表すと右のようになります。

7. フィボナッチ数列 [数学 B]

レオナルド・フィボナッチは次の問題を考案した数列です。高校の数学の教科書でも、フィボナッチ数列は数学 B で取り上げられています。さらにこのフィボナッチ数は、黄金比と関係があります。黄金比については、数学 I で取り上げられています。異なった章で学ぶことが実は関係があるということを発見するとなんとも嬉しくなります。ここでは、フィボナッチ数と黄金比の関係を軽く理解しましょう。

ノック 48 本目

1 つがいの兎は、産まれて 2 か月後から毎月 1 つがいずつの兎を産む。
兎が死なずずっと生み続ける。この条件の下で、産まれたばかりの 1 つがいの兎は 1 年の間に何つがいの兎になるか？



6 カ月目までを図示し、1 から 6 カ月の兎のつがいの数を数えると、1、 1 、2、 3、5、8 となる。この数列をフィボナッチ数 (Fibonacci sequence) (F_n) といい、次の式で定義される。このフィボナッチ数を計算しよう。

$F_0 = 0,$

$F_1 = 1,$

$F_2 = F_0 + F_1=0+1=1$

$F_3 = F_1 + F_2=1+1=2$

$F_4 = F_2 + F_3=1+2=3$

$F_5 = F_3 + F_4=2+3=5$

…

$F_{n+2} = F_n + F_{n+1}$ (n ≥ 0)

これをプログラミングしてみよう。

RK48.R

```
1  ns<-12
2  ns<-ns+1
3  Fib<-rep(0,ns)
4  Fib<-c(0,1)
5  for(i in 3:ns){
6    Fib[i]<-Fib[i-1]+Fib[i-2]
7  }
8  Fib
9  
10 plot(1:ns,Fib,xlab="i",ylab="Fibonacci sequence")
```

```
> Fib
 [1]   0   1   1   2   3   5   8  13  21  34  55  89 144
```

はじめを 0 カ月としているので、Fib[13]が 12 か月後の兎のつがいの数 144 となる。

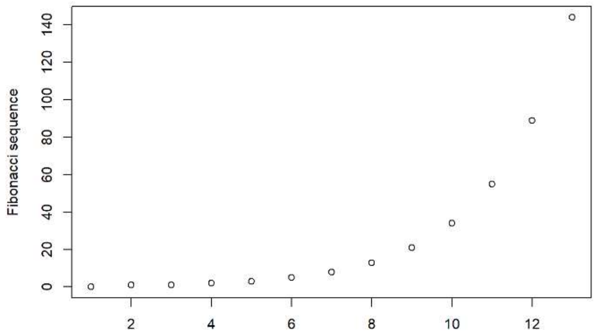
1 行目：ns<-12

2 行目：ns に 1 加えて 13 となっている。ここで、0 番目のフィボナッチ数 Fib[1]、12 番目を Fib[13]とした。

3 行目：Fib[1]から Fib[ns]のベクトルを定義する。とりあえず 0 が入っています。

4 行目：Fib[1]に 0 から Fib[2]に 1 を代入する。

5-6 行目：i を順次 3,4,⋯,ns として、Fib[3]、Fib[4]、⋯、Fib[ns]と求めます。



ノック 49 本目

i 番目のフィボナッチ数を F_i としよう

$$r_i = \frac{F_i}{F_{i-1}}$$

について、i を大きくすると黄金比に近づくという。そこで、

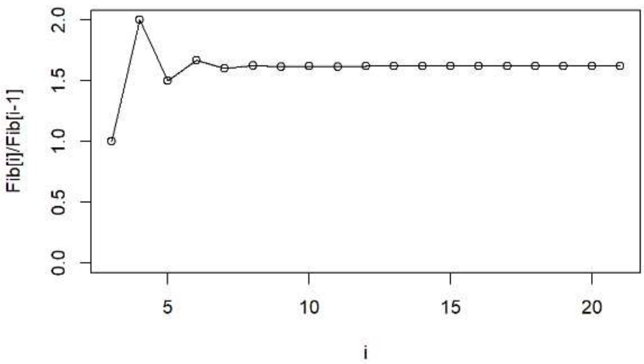
i=1,2,⋯,100 を横軸に、r_iを縦軸に図に表してみよう。

ちなみに、黄金比は、$1:\frac{1+\sqrt{5}}{2} \cong 1:1.618\ldots$で、「人間が最も美しいと感じる比率」なのだそうです。フィボナッチ数と黄金比の関係を図示しよう。

RK49.R

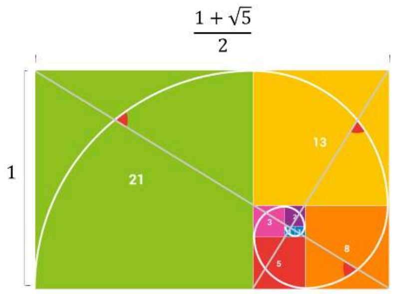
```
ns<-20
ns<-ns+1
Fib<-rep(0,ns)
Fib<-c(0,1)
for(i in 3:ns){
  Fib[i]<-Fib[i-1]+Fib[i-2]
}

```

```
r<-rep(0,ns-1)
for(i in 1:(ns-1)){
    r[i]<-Fib[i+1]/Fib[i]
}

plot(1:(ns-1),r,xlab="i",typ="b",ylab="Golden ratio")

r[ns-1]
(1+sqrt(5))/2
```




```
> r[ns]
[1] 1.618034
> (1+sqrt(5))/2
[1] 1.618034
```

ちなみに、黄金比はオウムガイの形とも関係あるのだそうです。



たねあかし

フィボナッチ数列とは

$$F_0 = 0$$
$$F_1 = 1$$
$$F_{n+2} = F_n + F_{n+1}$$

です。これは実は、

$$F_n = \frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right\}$$

と書けます。

$$F_0=\frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^0-\left(\frac{1-\sqrt{5}}{2}\right)^0\right\}=0$$

$$F_1=\frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^1-\left(\frac{1-\sqrt{5}}{2}\right)^1\right\}=\frac{1}{\sqrt{5}}\left\{\frac{1+\sqrt{5}-1+\sqrt{5}}{2}\right\}=\frac{1}{\sqrt{5}}\left\{\frac{2\sqrt{5}}{2}\right\}=1$$

$$F_2=\frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^2-\left(\frac{1-\sqrt{5}}{2}\right)^2\right\}=\frac{1}{\sqrt{5}}\left\{\frac{6+2\sqrt{5}-6+2\sqrt{5}}{4}\right\}=\frac{1}{\sqrt{5}}\left\{\frac{4\sqrt{5}}{4}\right\}=1$$

$$F_3=\frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^3-\left(\frac{1-\sqrt{5}}{2}\right)^3\right\}=\frac{1}{\sqrt{5}}\left\{\frac{16+8\sqrt{5}-16+8\sqrt{5}}{8}\right\}=\frac{1}{\sqrt{5}}\left\{\frac{8\sqrt{5}}{4}\right\}=2$$

…

では

$$F_k+F_{k+1}=\frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^k-\left(\frac{1-\sqrt{5}}{2}\right)^k\right\}=\frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^{k+1}-\left(\frac{1-\sqrt{5}}{2}\right)^{k+1}\right\}$$

$$=\frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^k\left(1+\frac{1+\sqrt{5}}{2}\right)-\left(\frac{1-\sqrt{5}}{2}\right)^k\left(1-\frac{1-\sqrt{5}}{2}\right)\right\}$$

$$=\frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^k\left(\frac{3+\sqrt{5}}{2}\right)-\left(\frac{1-\sqrt{5}}{2}\right)^k\left(\frac{3-\sqrt{5}}{2}\right)\right\}$$

ここで、

$$\left(\frac{1+\sqrt{5}}{2}\right)^2=\frac{6+2\sqrt{5}}{4}=\frac{3+\sqrt{5}}{2}$$

$$\left(\frac{1-\sqrt{5}}{2}\right)^2=\frac{6-2\sqrt{5}}{4}=\frac{3-\sqrt{5}}{2}$$

であることを活用すると、

$$F_k+F_{k+1}=\frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^k\left(\frac{3+\sqrt{5}}{2}\right)-\left(\frac{1-\sqrt{5}}{2}\right)^k\left(\frac{3-\sqrt{5}}{2}\right)\right\}$$

$$=\frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^k\left(\frac{1+\sqrt{5}}{2}\right)^2-\left(\frac{1-\sqrt{5}}{2}\right)^k\left(\frac{1-\sqrt{5}}{2}\right)^2\right\}=F_{k+2}$$

ということで、

$$F_k+F_{k+1}=F_{k+2}$$

が成り立っています。

では、

$$\lim_{n\to\infty}\frac{F_{n+1}}{F_n}$$

としたときの極限値を求めてみましょう。式を一生懸命変形すると、

$$\frac{F_{n+1}}{F_n}=\frac{\left(\frac{1+\sqrt{5}}{2}\right)-\left(\frac{1-\sqrt{5}}{2}\right)\left(\frac{1-\sqrt{5}}{1+\sqrt{5}}\right)^n}{1-\left(\frac{1-\sqrt{5}}{1+\sqrt{5}}\right)^n}$$

n→∞とすると、$\left(\frac{1-\sqrt{5}}{1+\sqrt{5}}\right)^n$→0 となるので、

$$\lim_{n\to\infty}\frac{F_{n+1}}{F_n}=\frac{1+\sqrt{5}}{2}$$

となります。これが黄金比となる訳だ、なるほど！

8. 素数 [数学 A]

素数とは、1 とそれ自身以外に正の約数をもたない自然数です。NEXT 数学 A「第 3 章 数学と人間の活動」でも取り上げられています。素数の分布に関する法則としては「リーマン予想」というのがあって、1859 年にドイツの数学者ベルンハルト・リーマン（1826～1866）によって提唱されました。その予想の具体的な内容は大変難しいのでここでは割愛しますが、一見ランダムに見える素数のすべてに共通する秩序があることになるのだそうです。リーマン予想の証明は現在も未解決であり、アメリカのクレイ数学研究所によって 100 万ドル(1 億 4 千万円)の懸賞金がかけられているのだそうで、相当、数学者が討ち死にしたのだろうと予想がつきます。リーマン予想と同じく、これまでに例外は見つかっていないものの、正しいことが証明されていない素数に関する「法則」として「4 以上の偶数はすべて、2 つの素数の足し算で表せる」というのもありますが、ここでは深入りするのはやめましょう。素数を求める単純なプログラムを作ってみましょう。

ノック 50 本目

RK50.R は、ある自然数について素数かどうか判定するプログラムを作成しよう。

いま、ある自然数を NN としよう。素数の定義から 2,3,…,NN-1 で割り算をしたときの余りが 0 とならなければ素数である。プログラムでは、79190 が素数であるかを判定する。

NN を 79190 とすると

```
[1] "79190 is NOT Prime Number yo!"
```

と出力され素数(Prime Number)でないことがわかり、約数が表示される。

```
> which(disc==1)
[1]     2     5    10  7919 15838 39595
```

では、NN を 7919 としてみよう。すると

```
[1] "7919 is Prime Number yo!"
```

となり素数とわかります。

```
> which(disc==1)
integer(0)
```

となり 1 と自分自身以外の約数はないことがわかります。

RK50.R

```
NN<-79190
disc<-rep(0,NN)

```

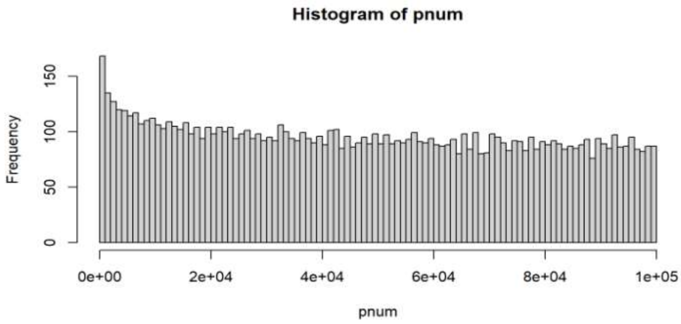
```
for(i in 2:(NN-1)){

  if(NN%%i==0){
    disc[i]<-1
  }
}

if(sum(disc)==0){
  pp<-paste(NN,"is Prime Number yo!")
  print(pp)
}else{
  pp<-paste(NN,"is NOT Prime Number yo!")
  print(pp)
}

which(disc==1)
```

ノック 51 本目

100000 の中の素数を求めて、1000 ごとに区切りでヒストグラムをつくってみよう。

RK51.R

```
maxN<-100000

pnum<-c(2)
np<-1
for(i in 3:maxN){

  ncount<-0

  for(k in 1:np){
    if(i%%pnum[k]!=0){
      ncount<-ncount+1
    }else{
      break
    }

  }

  if(ncount==np){
      pnum<-c(pnum,i)
      np<-np+1
  }
}

ns<-length(pnum)
his<-hist(pnum,seq(0,maxN,1000))
his$count
```



1000 ずつに区切って素数の数を求めると徐々に減っているようには見えるが…。

自然界の素数の話題

素数ゼミ

北アメリカで、17 年あるいは 13 年という素数を年周期で大発生をするセミがいる。セブテンデシム(*Magicicada septendecim*)、カッシーニ(*Magicicada cassini*)、セブテンデキュラ(*Magicicada septendecula*)は 17 年周期で大発生する。一方、トレデシム(*Magicicada tredecim*)、ネオトレデシム(*Magicicada neotredecim*）、トレでキュラ(*Magicicada tredecula*)は 13 年周期で大発生する。この周期性は、それぞれの種の交雑を防ぐことと関連するらしい。生物の発生周期が素数で説明できる？(吉村仁著、「17 年と 13 年だけ大発生？：素数ゼミの秘密に迫る！」(サイエンス・アイ新書)。

最大公約数(greatest common measure)

ユークリッドの互除法

二つの整数（とりあえず共に正)としよう。

まず、二つの整数の大きい方を r_0、残りを r_1 とする。

(第 1 回)　$r_0 \div r_1 = p_1$ 余り $r_2 \neq 0$

(第 2 回)　$r_1 \div r_2 = p_2$ 余り $r_3 \neq 0$

(第 3 回)　$r_2 \div r_3 = p_3$ 余り $r_4 \neq 0$

：

のように割り算の余り r_n-1 が 0 でなければ、その余り r_{n-1} で除数 r_n を割っていく ことを繰り返していくと，あるところで必ず

(第 k 回)　$r_{k-1} \div r_k = p_k$　余り $r_{k+1} \neq 0$

(第 k+1 回）$r_k \div r_{k+1} = p_{k+1}$ 余り $r_{k+2} = 0$

と割りきれるので，このとき

最大公約数は、$gcd(A,B) = r_{k+1}$

つまり、$r_{k+2}=0$ となったときの、r_{k+1} が最大公約数となる。

これをプログラミングしよう。

ノック 52 本目

6006 と 391391 の最大公約数を求めよう。

RK52.R

```
x1<-6006
x2<-391391
r0<-max(x1,x2)
r1<-min(x1,x2)
r2<-r0%%r1
bingo<-r1

while(r2!=0){
  r3<-r1%%r2
  paste(r3,r2)
  print(paste("[r1%%r2]=",r3,"[r2]=",r2))
  bingo<-r2
  r1<-r2
  r2<-r3
}
bingo
```

```
> bingo
[1] 1001
```

最小公約数(largest common multiple)を求めよう。

ノック 53 本目

ノック 52 本目で作成した最小公倍数をもとにプログラムをつくってみよう。

たねあかし

aとbの最大公約数G、bとrの最大公約数をgとしたときに、$G=g$となります。まずこれを示しましょう。

aとbで割ったときの商をq余りをrとすると

$$a = bq + r \qquad (1)$$

と表すことができます。

aとbの最大公約数はG、bとrの最大公約数はgであるので

$$a = Ga' \qquad (2)$$

$$b = Gb' = gb'' \qquad (3)$$

$$r = gr'' \qquad (4)$$

とあらわすことができます。ここで、a', b', b'', r''は自然数です。

(1)が成り立つので、これに(2)と(3)式をもとに

$$a = bq + r = g(b''q + r'')$$

と変形できて、aはgで割り切れることがわかります。またbはもともとgで割り切れるので、gはaとbの公約数です。Gはaとbとの最大公約数なので

$$g \leq G \qquad (5)$$

が成り立ちます。

次に(1)を変形して

$$r = a - bq = G(a' - b'q)$$

と表せるので、r はGでも割り切れます。つまりGはbとrの約数となります。

bとrの最大公約数は g なので、

$$G \leq \mathrm{g} \tag{6}$$

が成り立ちます。

そこで、(5)と(6)の条件を満たすのは、

$$G = \mathrm{g} \tag{7}$$

となります。

これで aとbの最大公約数Gとbとrの最大公約数をgとしたときに、$G = g$となることが示されました。

ということは、

$$a_1 = b_1 q_1 + r_1$$

a_1をb_1で割ったときの余りをr_1を求めると、$a_1 > b_1$ 、 $a_1 > r_1$となります。ここで$r_1 = 0$となれば、q_1が最大公約数となります。

$r_1 = 0$ でないときには、b_1をr_1で割った余りr_2を求めます。

$$b_1 = r_1 q_2 + r_2$$

ここで$r_2 = 0$ となれば、q_2が最大公約数となります。このように順次r_iが 0 になるまで割り算をしていき、$r_i = 0$となれば、r_{i-1}が最大公約数となります。

つまり

$$b_1 > r_1 > r_2 > \cdots > r_n > r_{n+1} = 0$$

と順次 r_iを求めていきr_{n+1}が 0 となったときのr_nが最大公約数となります。

ノック 54 本目

37037037 と 86419753 の最大公約数と最小公倍数を求めよう。

ノック 55 本目

148005 と 376805 の最大公約数を求めよう。

ノック 56 本目

6006 と 391391 の最小公倍数を求めよう。

9. 無限数列の和 [数学 III]

無限等比級数などさまざまな規則性のある級数の総和が意外にもπに近づいたり、整数になったりと思いがけない数値と等しくなります。この章では、規則性のある数列の和の収束する値をプログラミングを通して観察してみましょう。数学 B「数列」の発展形だと思えば、たいして難しくありません。いえいえ、証明しろと言われれば結構むずかしいものもありますが、計算機実験でなるほどと思うには面白い課題だと思います。

例えば、初項が 1 の等比級数の和を

$$S_n = 1 + r + r^2 + \cdots + r^{n-1} = \sum_{k=0}^{n-1} r^k$$

とします。この等比級数の和をもとに

$$rS_n = r + r^2 + \cdots + r^{n-1} + r^n$$

をつくり、上式から下式を引くと、

$$S_n = \frac{1-r^n}{1-r}$$

が得られます。ここで、$|r| < 1$のとき、n をどんどん大きく(n→ ∞)すれば、$r^n \to 0$となるので、等比級数の和S_nは$\frac{1}{1-r}$に近づきます。この r を 1/2 とすると

$$S_n = \left(\frac{1}{2}\right)^0 + \cdots + \left(\frac{1}{2}\right)^n$$

の n を無限大におおきくすると 2 に近づきます。これは

$$S_n = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots.$$

と同じ式で、なんとも分母が 2 倍ずつ増えていて規則性のある級数です。また$\frac{1}{1-r} = \frac{1}{1-\frac{1}{2}} = 2$となるので、$S_n$は 2 に近づきます。以下ではいろいろな級数の例を紹介します。終息する値はなんだろうと楽しんでください！

$\sum_{k=0}^{\infty} r^k = \frac{1}{1-r}$ ただし、$|r| < 1$

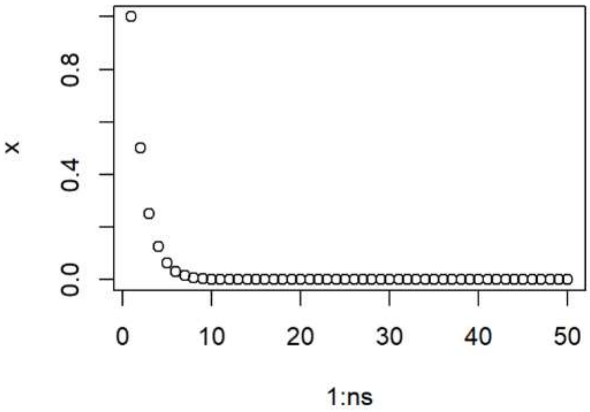
ということを、高校数学 B[第 1 章　数列]で習います。これを図に表すとどうなるのでしょうか？いろいろな規則性のある数列について図に表しながら考えてみましょう。

ノック 57 本目

$r=\frac{1}{2}$としたときの級数を$\left(\frac{1}{2}\right)^0$．$\left(\frac{1}{2}\right)^1$．$\left(\frac{1}{2}\right)^2\cdots,\left(\frac{1}{2}\right)^{49}$を縦軸、横軸を 0,1,⋯,49 として散布図をつくってみよう。

RK57.R

```
ns<-50
r<- 1/2
x<-rep(0,ns)
for(i in 1:ns){
  x[i]<-r^(i-1)
}
plot(1:ns,x)
```



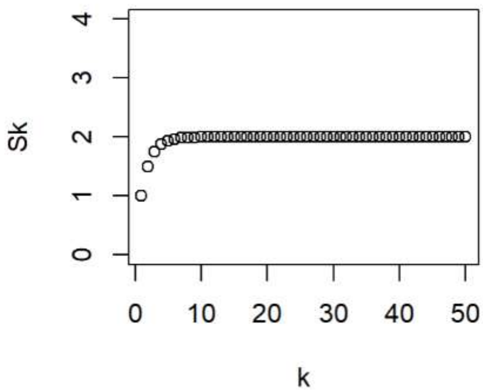
ノック 58 本目

$r=\frac{1}{2}$としたときの級数を$\left(\frac{1}{2}\right)^0$．$\left(\frac{1}{2}\right)^1$．$\left(\frac{1}{2}\right)^2\cdots,\left(\frac{1}{2}\right)^{49}$について、 $S_0=\left(\frac{1}{2}\right)^0$、$S_1=\left(\frac{1}{2}\right)^0+\left(\frac{1}{2}\right)^1$、$S_2=\left(\frac{1}{2}\right)^0+\left(\frac{1}{2}\right)^1+\left(\frac{1}{2}\right)^2$、$S_{50}=\left(\frac{1}{2}\right)^0+\cdots+\left(\frac{1}{2}\right)^{49}$ をもとめて、横軸を 0-49、縦軸に S_0、S_1、⋯、S_{49}の値をプロットしてみよう。

RK58.R

```
ns<-50
r<- 1/2
x<-rep(0,ns)

for(i in 1:ns){
```

```
  x[i]<-r^(i-1)
}
sx<-rep(0,ns)
for(k in 1:ns){
  sx[k]<-sum(x[1:k])
}
length(sx)
plot(0:49,sx,ylim=c(0,2*max(sx)),
     xlab="k",ylab="Sk")
sx[1:50]
```



$r = \frac{1}{2}$ のとき、

$$\sum_{k=0}^{n\infty} \left(\frac{1}{2}\right)^k = \frac{1}{1-\frac{1}{2}} = 2$$

となるはずで、実際に、S_{22}から 2 となった。なるほど！

```
> sx[1:50]
 [1] 1.000000 1.500000 1.750000 1.875000 1.937500 1.968750 1.984375 1.992188
 [9] 1.996094 1.998047 1.999023 1.999512 1.999756 1.999878 1.999939 1.999969
[17] 1.999985 1.999992 1.999996 1.999998 1.999999 2.000000 2.000000 2.000000
[25] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
[33] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
[41] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
[49] 2.000000 2.000000
```

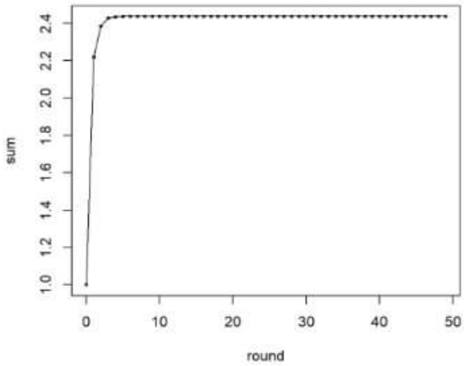
ノック 59 本目

級数を 1、$\frac{1}{1}$、$\frac{1}{1\cdot 2}$、$\frac{1}{1\cdot 2\cdot 3}\cdots, \frac{1}{1\cdot 2\cdot 3\cdot\ldots\cdot n}$について、
$S_0 = 1$、$S_1 = 1 + \frac{1}{1}$、$S_2 = 1 + \frac{1}{1} + \frac{1}{1\cdot 2}$、$S_3 = 1 + \frac{1}{1} + \frac{1}{1\cdot 2} + \frac{1}{1\cdot 2\cdot 3}$、$\cdots$、$S_{49} = \frac{1}{1} + \frac{1}{1\cdot 2} + \frac{1}{1\cdot 2\cdot 3} + \cdots + \frac{1}{1\cdot 2\cdot 3\cdot\ldots\cdot 49}$、
をもとめて、横軸を 0-49、縦軸に S_1、S_2、$\cdots$、S_{50} の値をプロットしてみよう。

RK59.R

```
ns<-50
a<-rep(0,ns)
a[1]<-1
sumv<-1
for(i in 1:ns){
  sumv<-sumv+1/factorial(i)                    #[3]
```

```
  a[i]<-sumv
}
plot(0:(ns-1),a,type="o",cex=0.5,xlab="round",ylab="sum")
length(a)
a[ns]
```




```
> length(a)
[1] 50
> a[ns]
[1] 2.718282
```

ネイピア数(数学定数の一つであり、自然対数の底)。

e = 2.71828 18284 59045 23536 02874 71352 …

に近づきます。ちなみにネイピア数の定義は、

$$e = \lim_{n\to\infty}(1+\frac{1}{n})^n$$

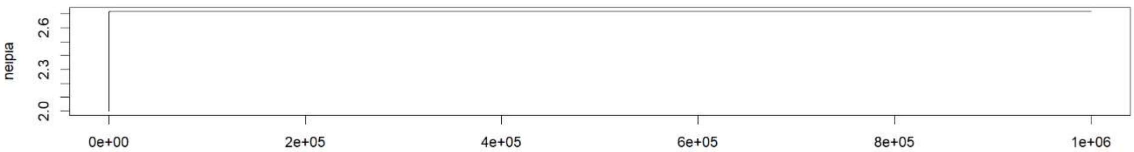
です。

実際にこの式により、n=1, 2, …,1000000 と$(1+\frac{1}{n})^n$を計算してみましょう(program A を実行してみましょう)。すると、n=1000000 では$(1+\frac{1}{n})^n$は、2.71828、ネイピア数と小数点 5 桁で一致します。また n=1,2,…,1000000 のときの$(1+\frac{1}{n})^n$の値は以下のようになります。

```
> exp(1)
[1] 2.718282
> neipia[n]
[1] 2.71828
```

program A

```
n<-1000000
neipia<-rep(0,n)
for(i in 1:n){
  neipia[i]<-(1+1/i)^i
}
plot(1:n,neipia,type='l')
exp(1)
neipia[n]
```



たねあかし

マクローリン展開という便利な方法があります。

$$f(x) = f(0) + \frac{1}{1!}f'(0)x + \frac{1}{2!}f^{(2)}(0)x^2 + \frac{1}{3!}f^{(3)}(0)x^3 + \cdots + \frac{1}{n!}f^{(n)}(0)x^n$$

と展開できます。$f^{(2)}(0)$、$f^{(3)}(0)$、…、$f^{(n)}(0)$はそれぞれ、$f(x)$の 2 階、3 階、…、n 階微分に 0 を代入した値です。

$$f(x) = x^3 + x^2 + x + 1$$

をマクローリン展開をしてみましょう。

$$f'(x) = 3x^2 + 2x + 1$$
$$f^{(2)}(x) = 6x + 2$$
$$f^{(3)}(x) = 6$$

となります。すると

$$f(0) = 1$$
$$f'(0) = 1$$
$$f^{(2)}(0) = 2$$
$$f^{(3)}(0) = 6$$

となると、

$$f(x) = 1 + \frac{1}{1!}1x + \frac{1}{2!}2x^2 + \frac{1}{3!}6x^3 = 1 + x + x^2 + x^3$$

と元の式が再現できます。

続いて

$$f(x) = e^x$$

とすると

$$e^x = 1 + \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \cdots + \frac{1}{n!}x^n$$

となるので、$x = 1$を代入すると、

$$\mathrm{e} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!}$$

となります。

ノック 60 本目

級数を$\frac{1}{1}$、 $-\frac{1}{2}$、 $\frac{1}{3}$、 $-\frac{1}{4}\cdots$, $(-1)^{n-1}\frac{1}{n}$について、

$S_1 = \frac{1}{1}$、$S_2 = \frac{1}{1} - \frac{1}{2}$、$S_3 = \frac{1}{1} - \frac{1}{2} + \frac{1}{3}$、$S_4 = \frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4}$、…、$S_n = \frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \cdots + (-1)^{n-1}\frac{1}{n}$、

を n=50 まで、横軸を 1-50、縦軸に S_1、S_2、…、S_{50} の値をプロットしてみよう。

実は、これは$log_e 2 = 0.69314718\ldots$に近づきます。はじめて少数第三桁まで一致するときの n を求めよう。

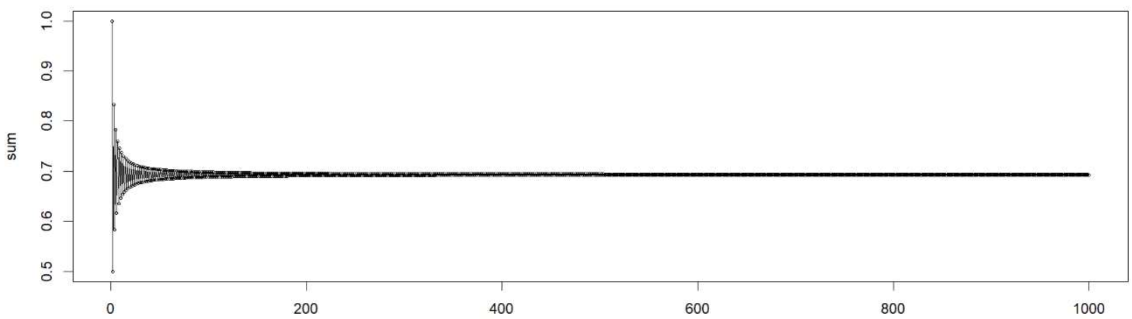
RK60.R

```
1  ns<-1000
2  S<-rep(0,ns)
3  sumv<-0
4  for(i in 1:ns){
5    addv<-((-1)^(i-1))/i
7    sumv<-sumv+addv
8    S[i]<-sumv
9  }
10 plot(1:ns,S,type="o",cex=0.5,xlab="round",ylab="sum")
11 
12 select<-which(0.693<S&S<0.694)
13 S[select[1]-2]
14 S[select[1]-1]
15 S[select[1]]
16 S[select[1]+1]
17 S[select[1]+2]
```

12 行目：S[1], S[2]…には 1 番目、2 番目,,,n 番目までの級数和が格納してあるので、級数和が$0.693 \le S[i] < 0.694$となる級数和を求めて、それが何番目であるかを which 関数で使って select に格納する。そこで select[1](=587)が最初にこの条件を満たします。これが「はじめて少数第三桁まで一致するときの n」と体操します。

13-17 行目：確認のため、select[1]の前後 2 個の級数和を出力した。

```
> S[select[1]-2]
[1] 0.6940012
> S[select[1]-1]
[1] 0.6922947
> S[select[1]]
[1] 0.6939982        ←ここではじめて 0.693 と小数三桁が一致する。
> S[select[1]+1]
[1] 0.6922976
> S[select[1]+2]
[1] 0.6939954
```



たねあかし

$log_e x$を$ln\,x$と簡略して書きます。$ln\,x$の微分は$\frac{1}{x}$となります。また、$ln(x+1)$の微分は$\frac{1}{(x+1)}$となります。となります。

$\frac{1}{x+1}$をマクローリン展開すると、

$$\frac{1}{x+1} = 1 - x + x^2 - x^3 + x^4 + \cdots$$

これを積分すると

$$ln(x+1) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \cdots$$

となります。そこで、$x = 1$を代入すると

$$ln(2) = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \cdots$$

となります。

ノック 61 本目

級数を$\frac{1}{1}$、 $-\frac{1}{3}$、 $\frac{1}{5}$、 $-\frac{1}{7}\cdots$, $(-1)^{2n-1}\frac{1}{2n-1}$について、

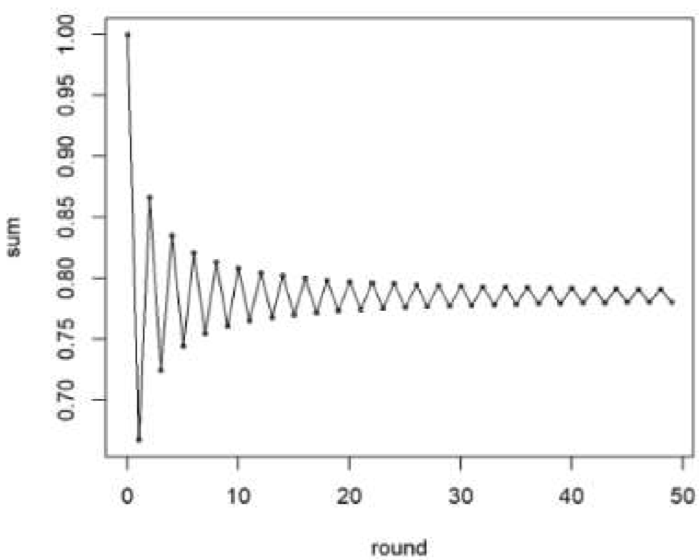
$S_1 = \frac{1}{1}$、$S_2 = \frac{1}{1} - \frac{1}{3}$、$S_3 = \frac{1}{1} - \frac{1}{3} + \frac{1}{5}$、$S_4 = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7}$、$\cdots$、$S_n = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots + (-1)^{2n-1}\frac{1}{2n-1}$、

を n=50 まで、横軸を 1-50、縦軸に S_1、S_2、$\cdots$、S_{50} の値をプロットしてみよう。

実は、これは$\pi/4 = 0.78539816\ldots$に近づきます。そこで少数第三桁まで一致するときの n を求めよう。

RK61.R

```
ns<-50
S<-rep(0,ns)
S[1]<-1
sumv<-0
for(i in 1:ns){
  sumv<-sumv+((-1)^(i-1))*1/(2*i-1)
  S[i]<-sumv
}

plot(1:ns-1,S,type="o",cex=0.5,xlab="round",ylab="sum")
S[ns]
pi/4
```



```
> a[ns]
[1] 0.7803987
> pi/4
[1] 0.7853982
```

たねあかし

$x = \tan y$とおいてxについて微分します。

$$1 = \frac{d}{dx}\tan y = \frac{1}{cos^2 y}\frac{dy}{dx}$$

すると

$$\frac{dy}{dx} = cos^2 y$$

$x = \tan y$ を用いて$cos^2 y$をxで表すと

$$cos^2 y = \frac{1}{1+x^2}$$

となります。

$\frac{1}{x+1}$をマクローリン展開すると、

$$\frac{1}{x+1} = 1 - x + x^2 - x^3 + x^4 - \cdots$$

これを利用すると、

$$\frac{1}{x^2+1} = 1 - x^2 + x^4 - x^6 + x^8 - \cdots$$

すると

$$\frac{dy}{dx} = \frac{1}{x^2+1} = 1 - x^2 + x^4 - x^6 + x^8 - \cdots$$

となると

$$\int \frac{dy}{dx} dx = \int \frac{1}{x^2+1} dx = \int (1 - x^2 + x^4 - x^6 + x^8 - \cdots) dx = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \frac{1}{9}x^9 - \cdots$$

$$\int \frac{dy}{dx} dx = \mathrm{y}$$

結局

$$\mathrm{y} = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \frac{1}{9}x^9 - \cdots$$

$x = 1$ とおくと、$x = \tan y$により、$\mathrm{y} = \frac{\pi}{4}$

よって

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots$$

となります。

ノック 62 本目

級数を$\frac{1}{1\cdot 3}$、 $\frac{1}{5\cdot 7}$、 $\frac{1}{9\cdot 11}$、 $\frac{1}{13\cdot 15}\cdots$, $\frac{1}{(4n-3)(4n-1)}$について、

$S_1 = \frac{1}{1\cdot 3}$、 $S_2 = \frac{1}{1\cdot 3} + \frac{1}{5\cdot 7}$、 $S_3 = \frac{1}{1\cdot 3} + \frac{1}{5\cdot 7} + \frac{1}{9\cdot 11}$、 $S_4 = \frac{1}{1\cdot 3} + \frac{1}{5\cdot 7} + \frac{1}{9\cdot 11} + \frac{1}{13\cdot 15}$、 $\cdots$、

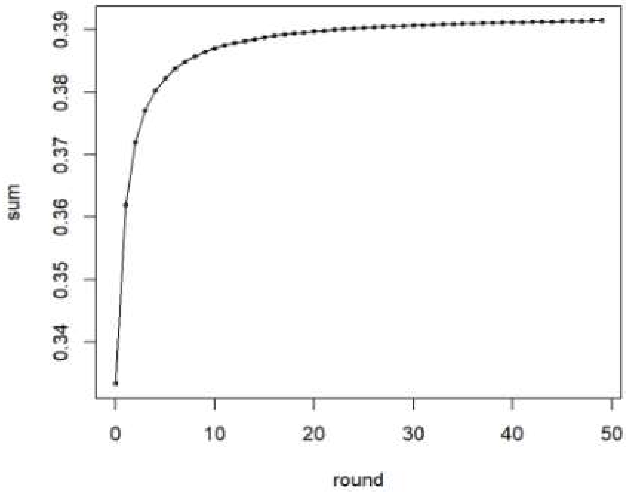
$S_n = \frac{1}{1\cdot 3} + \frac{1}{5\cdot 7} + \frac{1}{9\cdot 11} + \frac{1}{13\cdot 15} + \cdots + \frac{1}{(4n-3)(4n-1)}$、

を n=50 まで、横軸を 1-50、縦軸に S_1、S_2、…、S_{50}の値をプロットしてみよう。

実は、これは$\frac{\pi}{2\cdot 4} = 0.3926991\ldots$に近づきます。そこで少数第三桁まで一致するときの n を求めよう。

RK62.R

```
ns<-50
S<-rep(0,ns)
S[1]<-1
sumv<-0
for(i in 1:ns){
   sumv<-sumv+ 1/( (4*i-3)*(4*i-1) )
   S[i]<-sumv
}
plot(1:ns-1,a,type="o",cex=0.5,xlab="round",ylab="sum")
S[ns]
pi/(2*4)
```




```
> S[ns]
[1] 0.3914491
> pi/(2*4)
[1] 0.3926991
```

たねあかし

$\frac{1}{1\cdot 3}=\frac{1}{2}\left(\frac{1}{1}-\frac{1}{3}\right)$、

$\frac{1}{5\cdot 7}=\frac{1}{2}\left(\frac{1}{5}-\frac{1}{7}\right)$、

$\frac{1}{9\cdot 11}=\frac{1}{2}\left(\frac{1}{9}-\frac{1}{11}\right)$

…

を足すと

$\frac{1}{1\cdot 3}+\frac{1}{5\cdot 7}+\frac{1}{9\cdot 11}+\cdots=\frac{1}{2}\left(\frac{1}{1}-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\frac{1}{11}+\cdots\right)$

ノック 61 のたねあかしより

$$\frac{\pi}{4}=1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\cdots$$

を用いると

$$\frac{1}{1\cdot 3}+\frac{1}{5\cdot 7}+\frac{1}{9\cdot 11}+\cdots=\frac{1}{2}\frac{\pi}{4}$$

ノック 63 本目

(1) $S_n = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{n^2}$

(2) $S_n = \frac{1}{1^2} + \frac{1}{3^2} + \frac{1}{5^2} + \cdots + \frac{1}{(2n-1)^2}$

(3) $S_n = \frac{1}{2^2} + \frac{1}{4^2} + \frac{1}{6^2} + \cdots + \frac{1}{(2n)^2}$

について、n=50 まで、横軸を 1-50、縦軸に S_1、S_2、…、S_{50} の値をプロットしてみよう。

実は、これは上から順に

$$\frac{\pi^2}{6} = 1.625133$$

$$\frac{\pi^2}{8} = 1.233701$$

$$\frac{\pi^2}{24} = 0.4112335$$

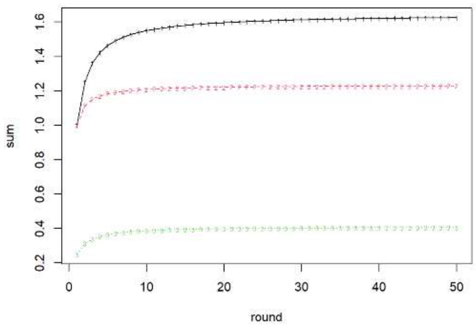
に近づきます。

RK63.R

```
1   ns<-50
2   S<-matrix(0,nrow=ns,ncol=3)
3   sumv1<-0
4   sumv2<-0
5   sumv3<-0
7
8   for(i in 1:ns){
9     sumv1<-sumv1+ 1/( i^2 )                    #[1]
10    S[i,1]<-sumv1
11
12    sumv2<-sumv2+ 1/( (2*i-1)^2 )              #[2]
13    S[i,2]<-sumv2
14
15    sumv3<-sumv3+ 1/( (2*i)^2 )                #[3]
16    S[i,3]<-sumv3
17  }
18  matplot(1:ns,S,type="o",cex=0.5,xlab="round",ylab="sum")
19  S[ns,1]
20  pi^2/6
21
22  S[ns,2]
23  pi^2/8
24
25  S[ns,3]
26  pi^2/24
```

1 行目：(1)、(2)、(3)のＳｎの値をそれぞれ、行列Ｓの 1-3 列目に格納した。行は i＝1,2,…ns です。

```
> S[ns,1]
[1] 1.625133
> pi^2/6
[1] 1.644934
>
> S[ns,2]
[1] 1.228701
> pi^2/8
```



```
[1] 1.233701
>
> S[ns,3]
[1] 0.4062832
> pi^2/24
[1] 0.4112335
```

たねあかしはノック 65 本目の説明を参考にしてください。

ノック 64 本目

$$S_n = \frac{1}{1^2} - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \cdots + (-1)^{n-1}\frac{1}{n^2}$$

について、n=50 まで、横軸を 1-50、縦軸に S_1、S_2、…、S_{50} の値をプロットしてみよう。

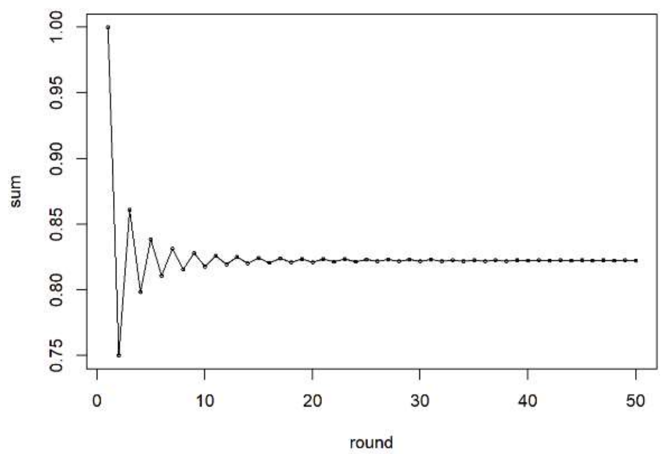
実は、 $\frac{\pi^2}{12} = 0.822467$

に近づきます。そこで少数第三桁まで一致するときの n を求めよう。

RK64.R

```
ns<-50
S<-rep(0,ns)
sumv<-0

for(i in 1:ns){
  sumv<-sumv+ ((-1)^(i-1))/(i^2)
  S[i]<-sumv
}

plot(1:ns,S,type="o",cex=0.5,xlab="round",ylab="sum")
S[ns]
pi^2/12
```

```
> S[ns]
[1] 0.822271
> pi^2/12
[1] 0.822467
```



ノック 65 本目

$$S_n = \frac{1}{1^2} - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \cdots + (-1)^{n-1}\frac{1}{n^2}$$

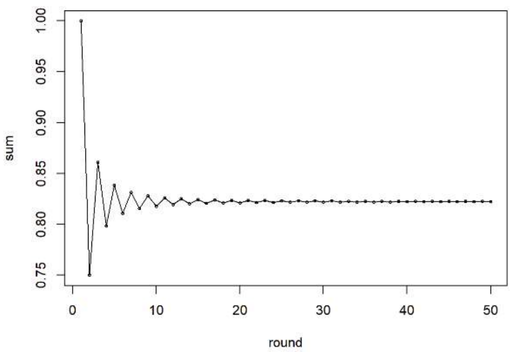
について、n=50 まで、横軸を 1-50、縦軸に S_1、S_2、⋯、S_{50} の値をプロットしてみよう。

実は、 $\frac{\pi^2}{12} = 0.822467$

に近づきます。

RK65.R

```
ns<-50
S<-rep(0,ns)
sumv<-0

for(i in 1:ns){
  sumv<-sumv+ ((-1)^(i-1))/(i^2)
  S[i]<-sumv
}

plot(1:ns,S,type="o",cex=0.5,xlab="round",ylab="sum")

S[ns]
pi^2/12
```




```
> S[ns]
[1] 0.822271
> pi^2/12
[1] 0.822467
```

たねあかし

(1) $\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots = \frac{\pi^2}{6}$ を示してみよう。

例えば、

$$(a+x)^n = a_0x + a_1x^1 + a_2x^2 + a_3x^3 + \cdots + a_nx^n$$

のように、展開できる。この発想を使うと、正し初項を 1 とすると

$$f(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + \cdots + x^n$$

は

$$g(x) = 1 + b_1x^1 + b_2x^2 + b_3x^3 + \cdots + b_nx^n$$

とおくと、これは

$$f(x) = (x - r_1)(x - r_2) \ldots (x - r_n)$$

と書くことができる。ここで、

$$g(x) = \left(1-\frac{x}{r_1}\right)\left(1-\frac{x}{r_2}\right)\dots\left(1-\frac{x}{r_n}\right)$$

とも書ける。

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

であるので

$$\frac{\sin x}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \cdots$$

一方で

$$\frac{\sin x}{x} = \left(1-\frac{x^2}{r_1}\right)\left(1-\frac{x^2}{r_2}\right)\dots$$

と書けます。

x^2の項に注目すると

$$-\frac{1}{3!} = -\left(\frac{1}{r_1}+\frac{1}{r_2}+\cdots\right)$$

ここで、$r_1 = \pi^2, r_2 = (2\pi)^2,\ r_2 = (3\pi)^2, \cdots$

すると

$$-\frac{1}{3!} = -\left(\frac{1}{r_1}+\frac{1}{r_2}+\cdots\right) = -\left(\frac{1}{\pi^2}+\frac{1}{(2\pi)^2}+\frac{1}{(3\pi)^2}\dots\right) = -\frac{1}{\pi^2}\left(1+\frac{1}{2^2}+\frac{1}{3^2}+\cdots\right)$$

つまり

$$\mathbf{1+\frac{1}{2^2}+\frac{1}{3^2}+\cdots=\frac{\pi^2}{3!}=\frac{\pi^2}{6}}$$

(2)　$\mathbf{1+\frac{1}{3^2}+\frac{1}{5^2}+\frac{1}{7^2}+\cdots=\frac{\pi^2}{8}}$　を示してみよう。

$1+\frac{1}{2^2}+\frac{1}{3^2}+\cdots$を偶数と奇数の項に分ける

$$1+\frac{1}{2^2}+\frac{1}{3^2}+\cdots = \left(1+\frac{1}{3^2}+\frac{1}{5^2}+\cdots\right)+\left(\frac{1}{2^2}+\frac{1}{4^2}+\cdots\right)$$
$$= \left(1+\frac{1}{3^2}+\frac{1}{5^2}+\cdots\right)+\frac{1}{2^2}\left(\frac{1}{1^2}+\frac{1}{2^2}+\frac{1}{3^2}\dots\right)$$

となり、

$$\frac{\pi^2}{6} = \left(1+\frac{1}{3^2}+\frac{1}{5^2}+\cdots\right)+\frac{1}{4}\frac{\pi^2}{6}$$

すなわち

$$\left(1+\frac{1}{3^2}+\frac{1}{5^2}+\cdots\right) = \frac{\pi^2}{6}-\frac{\pi^2}{24} = \frac{3\pi^2}{24} = \frac{\pi^2}{8}$$

(3)　$\mathbf{\frac{1}{2^2}+\frac{1}{4^2}+\frac{1}{6^2}+\cdots=\frac{\pi^2}{24}}$　を示してみよう。

(1)と(2)を利用して

$$1+\frac{1}{2^2}+\frac{1}{3^2}+\cdots = \left(\frac{1}{1^2}+\frac{1}{3^2}+\frac{1}{5^2}+\cdots\right)+\frac{1}{4}\left(\frac{1}{1^2}+\frac{1}{2^2}+\frac{1}{3^2}\dots\right)$$
$$\frac{3}{4}\left(\frac{1}{1^2}+\frac{1}{2^2}+\frac{1}{3^2}\dots\right) = \frac{\pi^2}{8}$$
$$\frac{1}{1^2}+\frac{1}{2^2}+\frac{1}{3^2}\dots = \frac{\pi^2}{6}$$

両辺を、$\frac{1}{2^2}$倍すると、

$$\frac{1}{2^2}+\frac{1}{4^2}+\frac{1}{6^2}\dots = \frac{\pi^2}{24}$$

(4)　$\mathbf{\frac{1}{1^2}-\frac{1}{2^2}+\frac{1}{3^2}-\frac{1}{34^2}\dots=\frac{\pi^2}{12}}$　を示してみよう。

(2)-(3)のを利用して

$$\left(1+\frac{1}{3^2}+\frac{1}{5^2}+\cdots\right)-\left(\frac{1}{2^2}+\frac{1}{4^2}+\frac{1}{6^2}+\cdots\right) = \frac{\pi^2}{8}-\frac{\pi^2}{24} = \frac{\pi^2}{12}$$

ノック 66 本目

$$a_1 = \frac{\sqrt{2}}{2},\ a_2 = \frac{\sqrt{2+\sqrt{2}}}{2},\ a_3 = \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2},\ \cdots$$

とする。

$$S_n = a_1 a_2 \cdots a_n$$

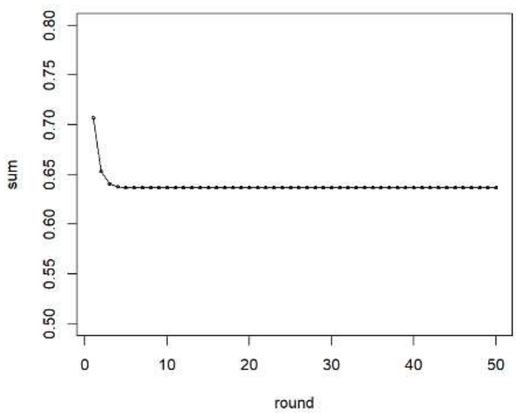
について、n=50 まで、横軸を 1-50、縦軸に S_1、S_2、…、S_{50} の値をプロットしてみよう。

実は、 $\frac{2}{\pi} = 0.6366198$

に近づきます。そこで小数第三桁まで一致するときの n を求めよう。

RK66.R

```
ns<-50
a<-rep(0,ns)

a[1]<-sqrt(2)
for(i in 2:ns){
   a[i]<-sqrt(2+a[i-1])
}

S<-rep(0,ns)
mulv<-1
for(i in 1:ns){
   mulv<-mulv*a[i]
   S[i]<-mulv/2^i
}

plot(1:ns,S,type="o",cex=0.5,xlab="round",ylab="sum",
       ylim=c(0.5,0.8))
S[ns]
2/pi
```




```
> S[ns]
[1] 0.6366198
> 2/pi
[1] 0.6366198
```

たねあかし

三角関数の２倍角の公式を使ってみよう。

$$sin2y = 2\cos y \sin y$$

この式に$2y = x$を代入すると、

$$\sin x = 2\cos\frac{x}{2}\sin\frac{x}{2}$$

さらに$\sin\frac{x}{2}$をじゃんじゃか半角にしていくと

$$\sin x = 2\cos\frac{x}{2}\left(2\cos\frac{x}{2^2}\sin\frac{x}{2^2}\right)=2\cos\frac{x}{2}\left(2\cos\frac{x}{2^2}\left(2\cos\frac{x}{2^3}\sin\frac{x}{2^3}\right)\right)$$
$$= 2^n\cos\frac{x}{2}\cos\frac{x}{2^2}\dots\cos\frac{x}{2^n}\sin\frac{x}{2^n}$$

$$\lim_{n\to\infty}\frac{\sin x}{x}=\lim_{n\to\infty}\frac{2^n\cos\frac{x}{2}\cos\frac{x}{2^2}\dots\cos\frac{x}{2^n}\sin\frac{x}{2^n}}{x} = \lim_{n\to\infty}\frac{2^2\cos\frac{x}{2}\cos\frac{x}{2^2}\dots\cos\frac{x}{2^n}\sin\frac{x}{2^n}}{\frac{x}{2^n}}$$
$$=\lim_{n\to\infty}\left(\cos\frac{x}{2}\cos\frac{x}{2^2}\dots\cos\frac{x}{2^n}\right)\frac{\sin\frac{x}{2^n}}{\frac{x}{2^n}}$$
$$= \cos\frac{x}{2}\cos\frac{x}{2^3}\dots\cos\frac{x}{2^n}$$

よって

$$\frac{\sin x}{x} = \cos\frac{x}{2}\cos\frac{x}{2^2}\dots\cos\frac{x}{2^n}$$

$x = \frac{\pi}{2}$を代入すると

$$\cos\frac{\pi}{2}=\frac{\sqrt{2}}{2}$$
$$\cos\frac{\pi}{2^2}=\frac{\sqrt{2+\sqrt{2}}}{2}$$
$$\cos\frac{\pi}{2^3}=\frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2}$$

…

$\cos\frac{\pi}{2^n}$の計算には、半角の公式 $\cos\frac{x}{2}=\frac{\sqrt{2+2\cos x}}{2}$ を使いましょう。

よって

$$\frac{2}{\pi} = \frac{\sqrt{2}}{2}\frac{\sqrt{2+\sqrt{2}}}{2}\frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2}\dots$$

ノック 67 本目

$$a_1 = \frac{2\cdot 2}{1\cdot 3},\ a_2 = \frac{4\cdot 4}{3\cdot 5},\ a_3 = \frac{6\cdot 6}{5\cdot 7},\ \cdots,\ a_n = \frac{(2n)\cdot(2n)}{(2n-1)(2n+1)},$$

とする。

$$S_n = a_1 a_2 \cdots a_n$$

について、n=50 まで、横軸を 1-50、縦軸に S_1、S_2、…、S_{50} の値をプロットしてみよう。

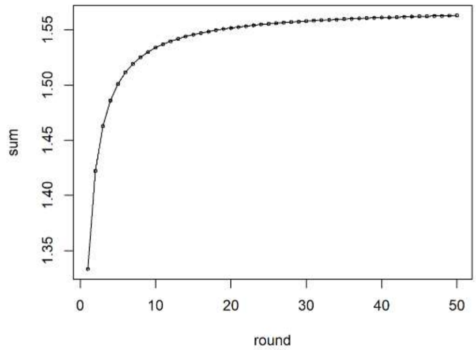
実は、 $\frac{\pi}{2}$ =1.570796

に近づきます。

RK67.R

```
ns<-50
a<-rep(0,ns)

for(i in 1:ns){
  a[i]<-2*i*2*i/( (2*i-1)*(2*i+1) )
}

S<-rep(0,ns)
mulv<-1
for(i in 1:ns){
  mulv<-mulv*a[i]
  S[i]<-mulv
}
plot(1:ns,a,type="o",cex=0.5,xlab="round",ylab="sum")

S[ns]
pi/2
```

```
> S[ns]
[1] 1.563039
> pi/2
[1] 1.570796
```



たねあかし

ノック 65 のたねあかしより

$$\frac{\sin x}{x} = \left(1 - \frac{x^2}{r_1}\right)\left(1 - \frac{x^2}{r_2}\right) \cdots$$

ここで、$r_1 = \pi^2, r_2 = (2\pi)^2,\ r_2 = (3\pi)^2, \cdots$

すなわち

$$\frac{\sin x}{x} = \left(1 - \frac{x^2}{\pi^2}\right)\left(1 - \frac{x^2}{(2\pi)^2}\right)\left(1 - \frac{x^2}{(3\pi)^2}\right) \cdots$$

$x = \frac{\pi}{2}$を代入すると

$$\left(1 - \frac{\left(\frac{\pi}{2}\right)^2}{\pi^2}\right) = \frac{3}{4} = \frac{1\cdot 3}{2\cdot 2}$$

$$\left(1 - \frac{\left(\frac{\pi}{4}\right)^2}{(2\pi)^2}\right) = \frac{15}{16} = \frac{3\cdot 5}{4\cdot 4}$$

$$\left(1 - \frac{\left(\frac{\pi}{4}\right)^2}{(3\pi)^2}\right) = \frac{35}{36} = \frac{5\cdot 7}{6\cdot 6}$$

…

$$\frac{2}{\pi} = \frac{1\cdot 3}{2\cdot 2}\frac{3\cdot 5}{4\cdot 4}\frac{5\cdot 7}{6\cdot 6} \cdots$$

よって

$$\frac{\pi}{2} = \frac{2\cdot 2}{1\cdot 3}\frac{4\cdot 4}{3\cdot 5}\frac{6\cdot 6}{5\cdot 7} \cdots$$

ノック 68 本目

$$S_n = \frac{3\cdot3}{2\cdot4}\cdot\frac{6\cdot6}{5\cdot7}\cdot\frac{9\cdot9}{8\cdot10}\cdots\frac{(3n)\cdot(3n)}{(3n-1)(3n+1)}$$
$$S_n = \frac{4\cdot4}{3\cdot5}\cdot\frac{8\cdot8}{7\cdot9}\cdot\frac{12\cdot12}{11\cdot13}\cdots\frac{(4n)\cdot(4n)}{(4n-1)(4n+1)}$$
$$S_n = \frac{6\cdot6}{5\cdot7}\cdot\frac{12\cdot12}{11\cdot13}\cdot\frac{18\cdot18}{17\cdot19}\cdots\frac{(6n)\cdot(6n)}{(6n-1)(6n+1)}$$

について、n=50 まで、横軸を 1-50、縦軸に S_1、S_2、…、S_{50} の値をプロットしてみよう。

上から順に、

$$\frac{2\pi}{3\sqrt{3}}=1.2092$$

$$\frac{\pi}{2\sqrt{2}}=1.110721$$

$$\frac{\pi}{3}=1.047198$$

に近づきます。そこで小数第三桁まで一致するときの n を求めよう。

これは各自でプログラミングしてみてください。

たねあかし

ノック 67 のたねあかしより

$x = \frac{\pi}{3}$、$\frac{\pi}{4}$、$\frac{\pi}{6}$

を代入する。

ノック 69 本目

$$\tan(x) = \frac{1}{5}$$
$$\tan(y) = \frac{1}{239}$$

となる x と y により、

$$4(4x - y) = \pi$$

となる。ここで

$$x = \tan^{-1}\left(\frac{1}{5}\right) = \arctan\left(\frac{1}{5}\right)$$
$$y = \tan^{-1}\left(\frac{1}{239}\right) = \arctan\left(\frac{1}{239}\right)$$

とあらわす。R では、atan(1/5), atan(1/239)とすると求められる。プログラムを書いてみよう。

RK69.R

```
4*(4*atan(1/5)-atan(1/239))
```

```
> 4*(4*atan(1/5)-atan(1/239))
[1] 3.141593
```

ノック 70 本目

$$S_n = \frac{1}{2^2}\cdot tan\left(\frac{\pi}{2^2}\right) + \frac{1}{2^3}\cdot tan\left(\frac{\pi}{2^3}\right) + \cdots + \frac{1}{2^n}\cdot tan\left(\frac{\pi}{2^n}\right)$$

について、n=50 まで、横軸を 1-50、縦軸に S_2、…、S_{50} の値をプロットしてみよう。これは

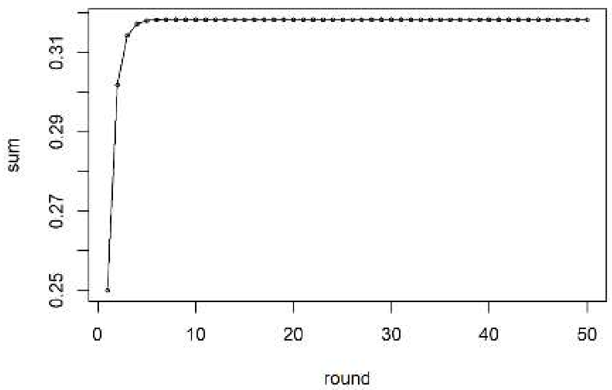
$$\frac{1}{\pi}=1.11072$$

に近づきます。そこで小数第三桁まで一致するときの n を求めよう。

RK70.R

```
ns<-50
tt<-rep(0,ns)

for(i in 1:ns){
  tt[i]<-1/(2^(i+1))*tan(pi/2^(i+1))
}

S<-rep(0,ns)
ss<-0
for(i in 1:ns){
  ss<-ss+tt[i]
  S[i]<-ss
}
plot(1:ns,S,type="o",cex=0.5,xlab="round",ylab="sum")
1/pi
S[ns]
```

```
> 1/pi
[1] 0.3183099
> S[ns]
[1] 0.3183099
```



たねあかし

$$\frac{1}{\tan A} = \frac{1-tan^2\left(\frac{A}{2}\right)}{2\tan\left(\frac{A}{2}\right)} = \frac{1}{2\tan\left(\frac{A}{2}\right)} - \frac{1}{2}\tan\left(\frac{A}{2}\right)$$

$$= \frac{1}{2^2\tan\left(\frac{A}{2^2}\right)} - \frac{1}{2^2}\tan\left(\frac{A}{2^2}\right) - \frac{1}{2}\tan\left(\frac{A}{2}\right)$$

……

$$= \frac{1}{2^n\tan\left(\frac{A}{2^n}\right)} - \frac{1}{2^n}\tan\left(\frac{A}{2^n}\right) - \cdots - \frac{1}{2}\tan\left(\frac{A}{2}\right)$$

ここで、$\frac{1}{2^n \tan\left(\frac{A}{2^n}\right)} = \frac{1}{A}\frac{A}{2^n}\frac{1}{\tan\left(\frac{A}{2^n}\right)} = \frac{1}{A}\frac{A}{2^n}\frac{\cos\left(\frac{A}{2^n}\right)}{\sin\left(\frac{A}{2^n}\right)} = \frac{1}{A}\frac{\cos\left(\frac{A}{2^n}\right)}{\frac{sin\left(\frac{A}{2^n}\right)}{\frac{A}{2^n}}}$

n→∞のとき、$\frac{\cos\left(\frac{A}{2^n}\right)}{\sin\left(\frac{A}{2^n}\right)}$→1、$\frac{sin\left(\frac{A}{2^n}\right)}{\frac{A}{2^n}}$→1

となるので、

$$\frac{1}{2^n \tan\left(\frac{A}{2^n}\right)} = \frac{1}{A}$$

$A = \frac{\pi}{4}$ とおくと$\frac{1}{2^n \tan\left(\frac{A}{2^n}\right)} = \frac{4}{\pi}$、また$\frac{1}{\tan\left(\frac{\pi}{4}\right)} = \tan\left(\frac{\pi}{4}\right)$

$$\tan\left(\frac{\pi}{4}\right) + \frac{1}{2}\tan\left(\frac{\pi}{8}\right) + \frac{1}{4}\tan\left(\frac{\pi}{16}\right) + \cdots + \frac{1}{2^n}\tan\left(\frac{\pi}{2^{n+2}}\right) = \frac{4}{\pi}$$

$$\frac{1}{4}\tan\left(\frac{\pi}{4}\right) + \frac{1}{8}\tan\left(\frac{\pi}{8}\right) + \frac{1}{16}\tan\left(\frac{\pi}{16}\right) + \cdots + \frac{1}{2^{n+2}}\tan\left(\frac{\pi}{2^{n+2}}\right) = \frac{1}{\pi}$$

ノック 71 本目

$$e^x = \left(1 + \frac{x}{n}\right)^n$$

一方で、

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

となることが知られている。では、x=1 としたときの

$$a_n = \left(1 + \frac{1}{n}\right)^n$$

と

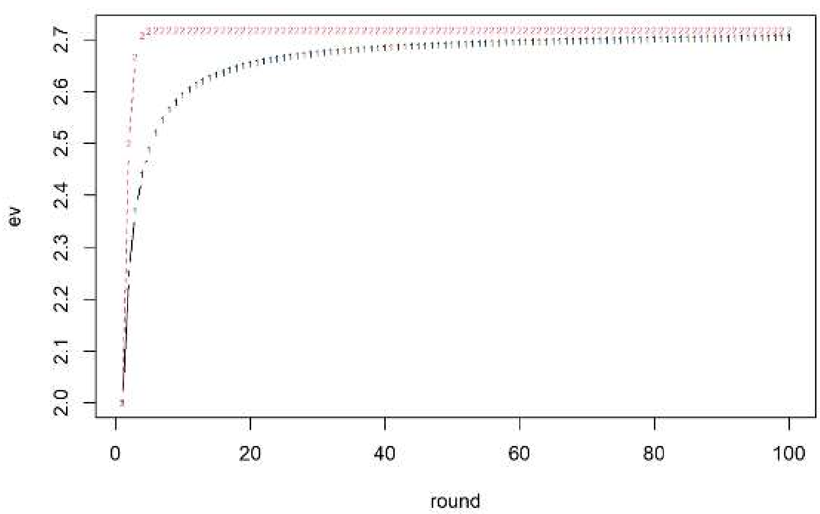
$$b_n = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!}$$

について、n=1,2,⋯,100 としたときのa_nとb_nを求めてみよう。すると、e=2.718282 に近づく。そこでa_nとb_nについて e=2.718282 と小数第三桁まで一致するときの n を求めよう。

RK71.R

```
ns<-100
a<-rep(0,ns)
b<-rep(0,ns)
sv<-1
for(i in 1:ns){
  a[i]<-(1+1/i)^i
  sv<-sv+1/factorial(i)
  b[i]<-sv
```

```
}
ev<-data.frame(a,b)
matplot(1:ns,ev,type="b",cex=0.5,xlab="round")
a[ns]
b[ns]
exp(1)
```




```
> a[ns]
[1] 2.704814
> b[ns]
[1] 2.718282
> exp(1)
[1] 2.718282
```

ノック 72 本目

$$b_n = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!}$$
$$c_n = \frac{1}{1!} + \frac{2}{2!} + \frac{3}{3!} + \cdots + \frac{n}{n!}$$

について、まあ、明らかに、b_nとc_nも n→∞のとき、e=2.718282 に近づく。

自力で作ってみよう！

たねあかし

$$c_n = \frac{1}{1!} + \frac{2}{2!} + \frac{3}{3!} + \cdots + \frac{n}{n!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!} = b_n$$

となります。

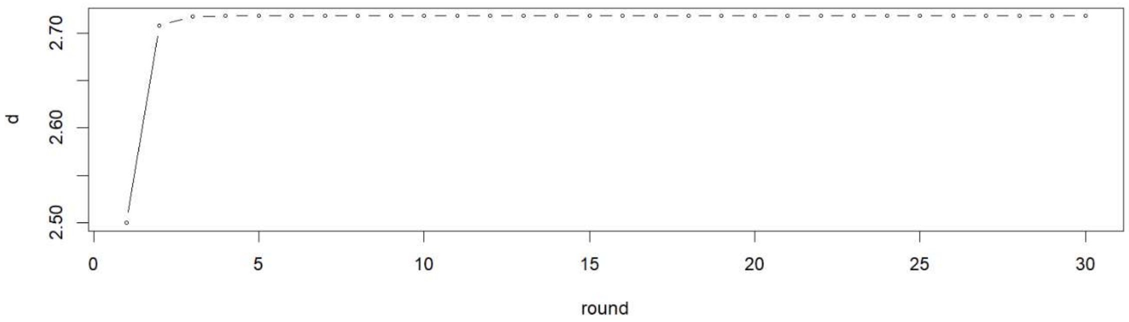
ノック 73 本目

$$d_n = 1 + \frac{3}{2!} + \frac{5}{4!} + \cdots + \frac{2n+1}{(2n)!} = 1 + \sum_{i=1}^{\infty} \frac{(2n+1)}{(2n)!}$$

についてd_nも n→∞のとき、e=2.718282 に近づく。

RK73.R

```
ns<-30
d<-rep(0,ns)

sv1<-1
for(i in 1:ns){
  sv1<-sv1+(2*i+1)/factorial(2*i)
  d[i]<-sv1
}
plot(1:ns,d,type="b",cex=0.5,xlab="round")
d[ns]
exp(1)
```




```
> a[ns]
[1] 2.718282
> c[ns]
[1] 2.718282
> exp(1)
[1] 2.718282
```

たねあかし

$$d_n = 1 + \frac{3}{2!} + \frac{5}{4!} + \cdots + \frac{2n+1}{(2n)!} = 1 + \sum_{i=1}^{\infty} \frac{(2n+1)}{(2n)!}$$

$$= 1 + \sum_{i=1}^{\infty} \left[\frac{2n}{(2n)!} + \frac{1}{(2n)!} \right]$$

$$= 1 + \sum_{i=1}^{\infty} \left[\frac{1}{(2n-1)!} + \frac{1}{(2n)!} \right]$$

$$= \left(\frac{1}{1!} + \frac{1}{3!} + \cdots\right) + \left(\frac{1}{2!} + \frac{1}{4!} + \cdots\right)$$
$$= \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \cdots$$
$$= e$$

ノック 74 本目

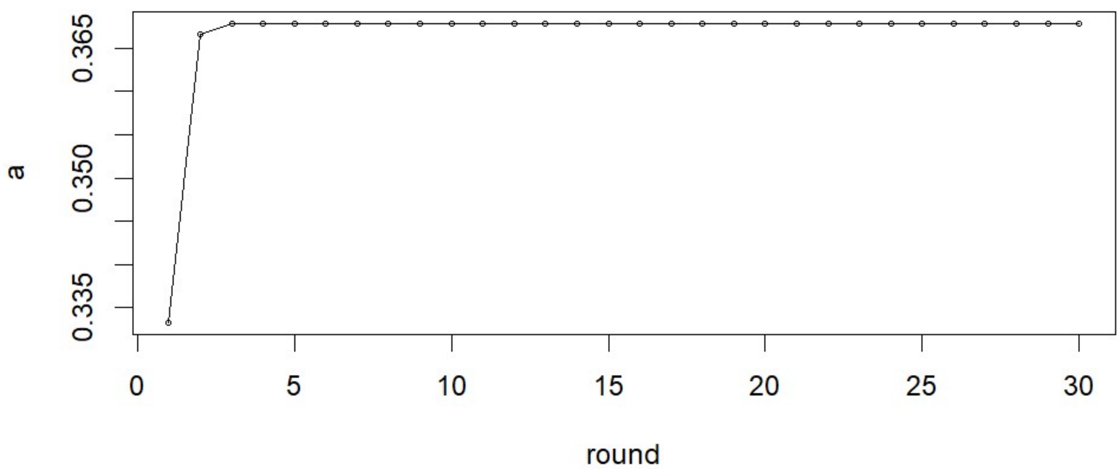
$$a_n = \frac{2}{3!} + \frac{4}{5!} + \frac{6}{7!} + \cdots + \frac{2n}{(2n+1)!}$$

についてa_nも n→∞のとき、1/e=1/2.718282=0.3678794 に近づく。

RK74.R

```
ns<-30
a<-rep(0,ns)

sv1<-0
for(i in 1:ns){
  sv1<-sv1+2*i/factorial(2*i+1)
  a[i]<-sv1
}
plot(1:ns,a,type="o",cex=0.5,xlab="round")
a[ns]
1/exp(1)
```

```
> a[ns]
[1] 0.3678794
> 1/exp(1)
[1] 0.3678794
```



たねあかし

$$a_n = \frac{2}{3!} + \frac{4}{5!} + \frac{6}{7!} + \cdots + \frac{2n}{(2n+1)!}$$

$$= \sum_{i=1}^{\infty} \left[\frac{2n}{(2n+1)!}\right]$$

$$= \sum_{i=1}^{\infty} \left[\frac{2n+1}{(2n+1)!} - \frac{1}{(2n+1)!}\right]$$

$$= \sum_{i=1}^{\infty} \left[\frac{1}{(2n)!} - \frac{1}{(2n+1)!}\right]$$

$$= \left(\frac{1}{2!} + \frac{1}{4!} + \cdots\right) - \left(\frac{1}{3!} + \frac{1}{5!} + \cdots\right)$$

$$= \left(\frac{1}{1!} + \frac{1}{2!} + \frac{1}{4!} + \cdots\right) - \left(\frac{1}{1!} + \frac{1}{3!} + \frac{1}{5!} + \cdots\right)$$

$$= \frac{1}{1!} - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} - \frac{1}{5!} + \cdots = \frac{1}{e}$$

ちなみに

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

について$x = -1$を代入すると

$$\frac{1}{e} = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{x^3}{3!} + \frac{1}{4!} - \frac{1}{5!} + \cdots$$

となります。

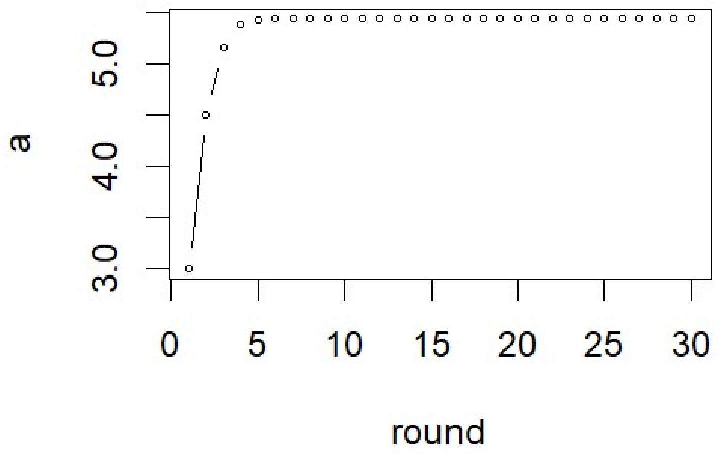
ノック 75 本目

$$a_n = \frac{1^2}{1!} + \frac{2^2}{2!} + \frac{3^2}{3!} + \cdots + \frac{n^2}{n!} = 1 + \frac{2}{1!} + \frac{3}{2!} + \cdots + \frac{n}{(n-1)!}$$

についてa_nも n→∞のとき、2e=2*2.718282=5.436564 に近づく。

RK75.R

```
ns<-30
a<-rep(0,ns)

sv<-1
for(i in 1:ns){
   sv<-sv+(i+1)/factorial(i)
   a[i]<-sv
}
plot(1:ns,a,type="b",cex=0.5,xlab="round")
a[ns]
2*exp(1)
```




```
> a[ns]
[1]  2.718282
> exp(1)
[1] 2.718282
```

たねあかし

$$\sum_{i=1}^{\infty} a_i = \frac{1^2}{1!}+\frac{2^2}{2!}+\frac{3^2}{3!}+\frac{3^2}{4!}+\cdots$$
$$= 1+\frac{2}{1!}+\frac{3}{2!}+\frac{4}{3!}+\frac{5}{4!}+\cdots$$
$$= 1+\sum_{i=1}^{\infty}\left[\frac{n+1}{n!}\right]$$
$$= 1+\sum_{i=1}^{\infty}\left[\frac{1}{(n-1)!}+\frac{1}{1!}\right]$$
$$= 1+e+(e-1)$$
$$= 2e$$

ノック 76 本目

$$a_n = \frac{1^2}{1!}+\frac{2^2}{2!}+\frac{3^2}{3!}+\cdots+\frac{n^2}{n!} = 1+\frac{2}{1!}+\frac{3}{2!}+\cdots+\frac{n}{(n-1)!}$$
$$b_n = \frac{1^2}{2!}+\frac{2^2}{3!}+\frac{3^2}{4!}+\cdots+\frac{n^2}{(n+1)!}$$

を比べてみよう。a_nは n→∞のとき、2e=2*2.718282=5.436564

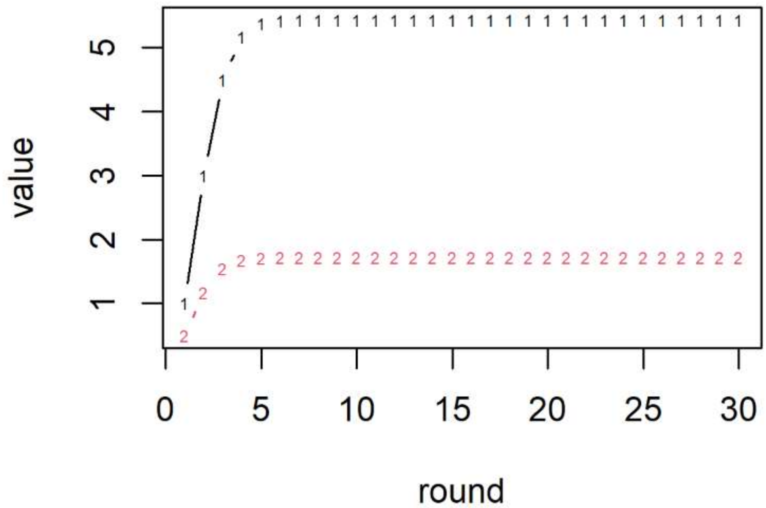
に近づく。

では、b_nは、e-1=2.718282-1=1.718282 に近づく。面白っ！

RK76.R

```
ns<-30
a<-rep(0,ns)
b<-rep(0,ns)
```

```
sv1<-0
sv2<-0
for(i in 1:ns){
  sv1<-sv1+(i^2)/factorial(i)
  a[i]<-sv1

  sv2<-sv2+(i^2)/factorial(i+1)
  b[i]<-sv2
}
mdata<-data.frame(a,b)
matplot(1:ns,mdata,type="b",cex=0.5,xlab="round",ylab="value")
a[ns]
b[ns]
2*exp(1)
exp(1)-1
```




```
> a[ns]
[1] 5.436564
> 2*exp(1)
[1] 5.436564
> b[ns]
[1] 1.718282
> exp(1)-1
[1] 1.718282
```

ノック 77 本目

$$a_n = \frac{k}{(k+1)^1} + \frac{k}{(k+1)^2} + \frac{k}{(k+1)^3} + \cdots + \frac{k}{(k+1)^n}$$

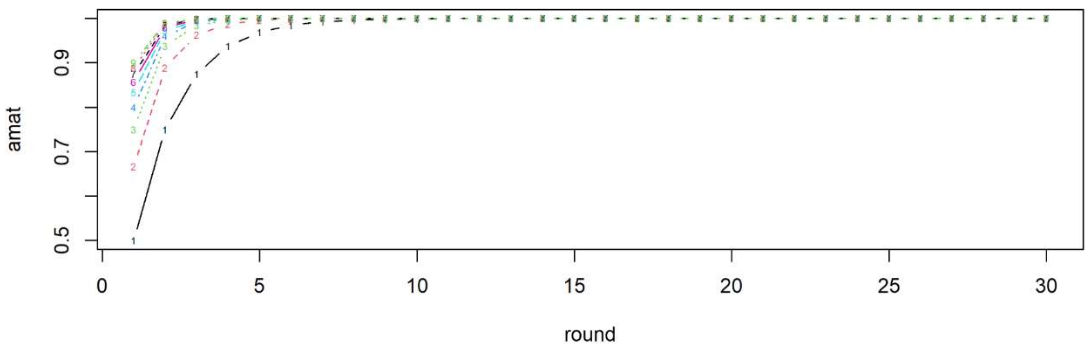
について、k=1,2,3,4,…,とどんな正の整数をいれても、1 に収束しまっせ！面白っ！

RK77.R

```
ns<-30
kmax<-9

```

```
amat<-matrix(0,nrow=ns,ncol=kmax)

for(k in 1:kmax){

  a<-rep(0,ns)
  sv<-0

  for(i in 1:ns){
    sv<-sv+k/((k+1)^i)
    a[i]<-sv
  }
  amat[,k]<-a
}

matplot(1:ns,amat,type="b",cex=0.5,xlab="round")
amat[ns,1:kmax]
```

amat 行列として、ns(i＝1,2,…,ns；ノック 77 本目の n と体操) 行、kmat (k=1,2,…,kmax; ノック 77 本目の k と体操)列を定義する。それぞれの k について 1 に収束すること視覚化した。




```
> amat[ns,1:kmax]
[1] 1 1 1 1 1 1 1 1 1
```

おわりに

　プログラミングを楽しんでもらえたでしょうか？高校の数学の課題をもとにちょっと発展させた課題を解くことはいい機会だと思います。いろいろな課題を解きながら、いろんな分野を楽しむことができれば、人生楽しくなるでしょう。でも、プログラミングにはまるとあっという間に時間が過ぎます。適に時間を考えて遊ぶのがいいでしょう。例えば、試験前にはやっちゃだめです。こっちのほうが面白いからついついはまってしまいます。でも気分転換にはもってこいの分野です。

アルバート・アインシュタイン
は言った。

数学の難問に遭遇したとしても
けっして案ずるには及ばない
私達の知の容量は
それよりおおきいからである。

ともあれ、お疲れさまでした。

付録.　R と RStudio のインストールと使い方

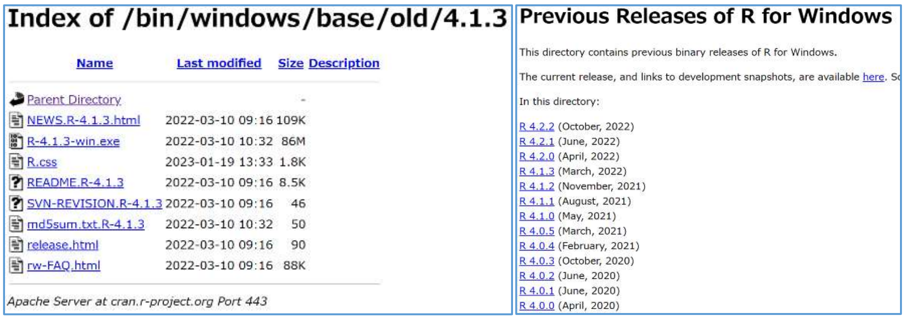
Rをデータ解析に活用するには、どうすればいいのだろうか？ RStudio でプログラムの作成から実行、プログラムファイルの格納までできます。そこでまず、R と RStudio のインストールを説明します。

R のインストール

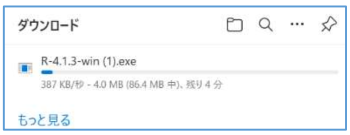
いろいろなバージョンの R をインストールすることができます。はじめに、ウインドウズの場合、

https://cran.r-project.org/bin/windows/base/old/

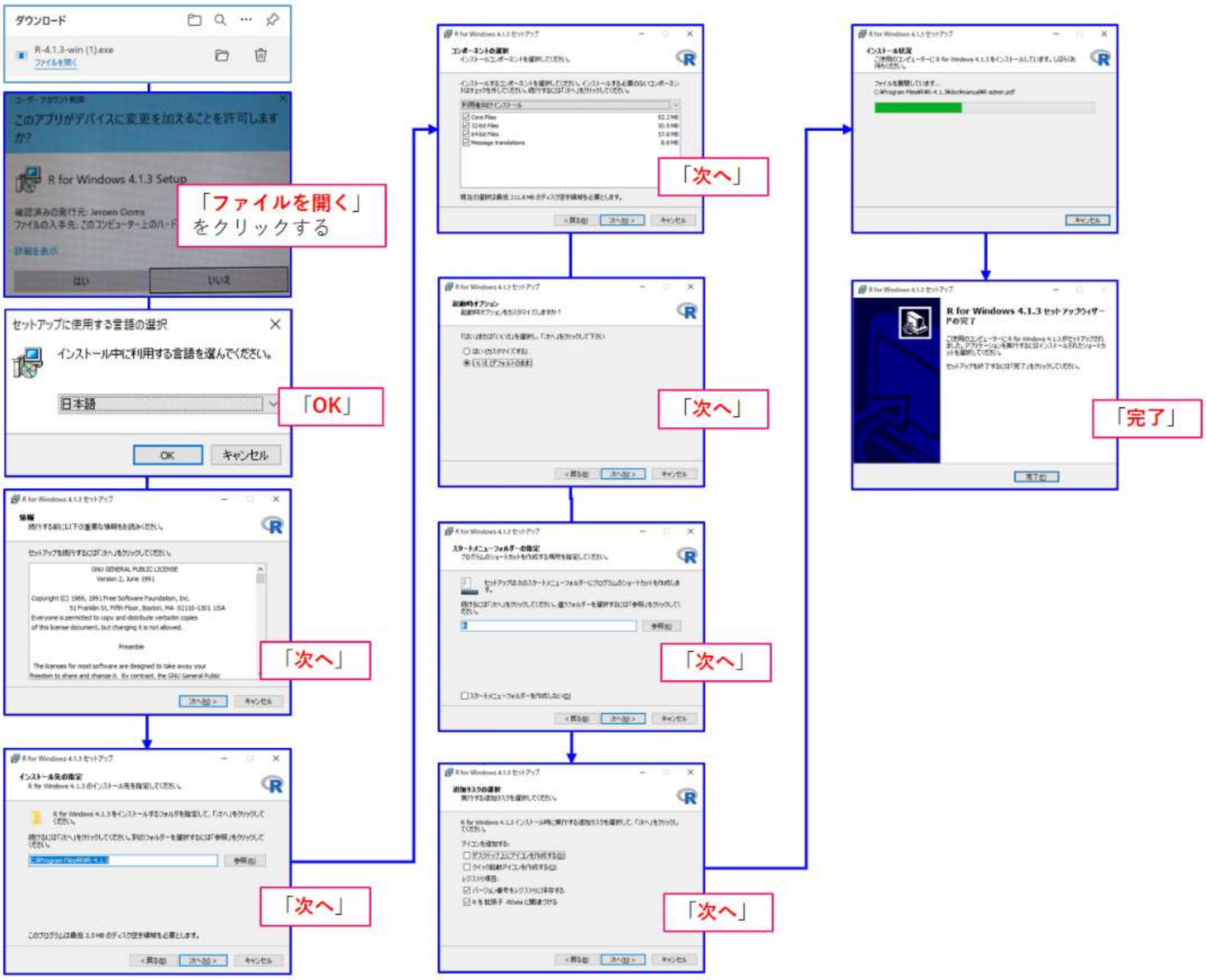
より、Previous Release of R for Windows として、以下の画面が表示されます。この中で新しいほどいいかというとそうではなく、新しいバージョンの方が、動作が不安定あるいはパッケージのインストールにてこずるなどの障害も出てきます。いまのところ R 4.1.3 がお勧めです。そこで R 4.1.3 をクリックしてみましょう。すると、Index of /bin/windows/base/old/4.1.3 のページが表示されます。



つづいて R-4.1.3-win.exe をクリックするとダウンロードが始まります。



exe ファイルのダウンロードが終わったところで、このファイルを実行すればインストールが完了です。以下にウインドウズにおけるインストールの一例を示します。



RStudio のインストール

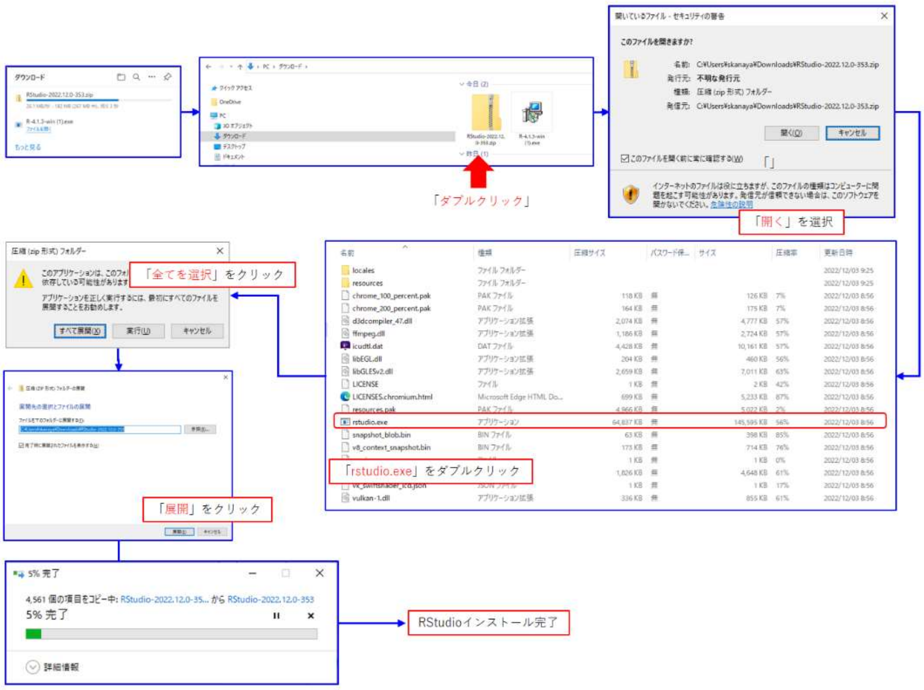
RStudio は

https://posit.co/download/rstudio-desktop/

からインストールすることができます。このサイトの 2.Install RStudio に Download サイトがリストアップされているので、ここからダウンロードします。仮にウインドウズの場合 Windows: 10/11 からダウンロードした例を以下に示します。

OS	Download	Size	SHA-256
Windows 10/11	RSTUDIO-2022.12.0-353.EXE	202.77 MB	FD8EA4B4
macOS 11+	RSTUDIO-2022.12.0-353.DMG	365.71 MB	FD4BEBB5
Ubuntu 18+/Debian 10+	RSTUDIO-2022.12.0-353-AMD64.DEB	131.20 MB	23CAE58F
Ubuntu 22	RSTUDIO-2022.12.0-353-AMD64.DEB	131.95 MB	8BC3F84D

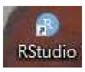
RSTUDIO-2022/12/0.353.EXE をクリックした後の操作例を以下に示します。



このようにして、RStudio のインストールが完了しました。

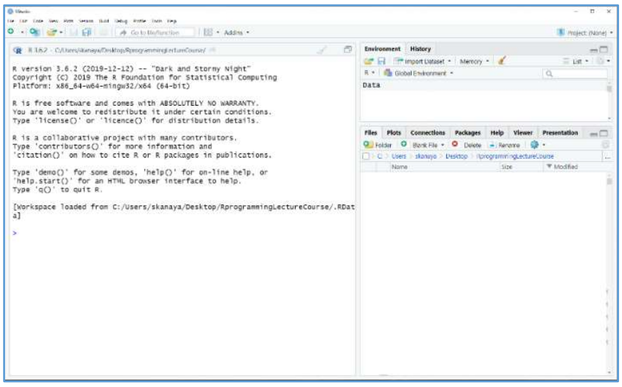
RStudio を立ち上げよう

　RStudio ファルダにある以下のアイコンをクリックすれば RStudio が立ち上がります。

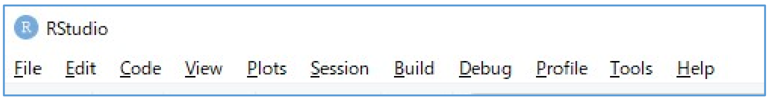


RStudio は R プログラミングをするときにしばしば使うので、パソコンの下側のアイコンに配置しておくと便利です。これは、上のアイコンをアイコンバーに移動することにより配置できます。

　では、RStudio　のアイコンをクリックして、実際に立ち上げてみましょう。すると、以下の画面が立ち上がります。

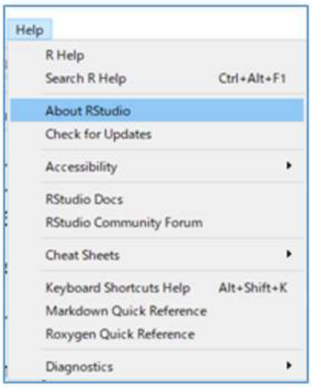


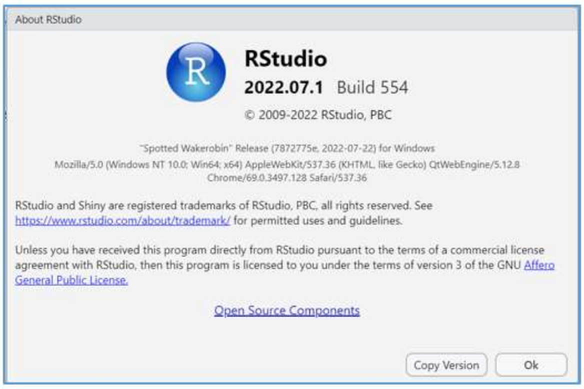
上段のメニューバーをみてみましょう。



RStudio のバージョンを確認する

RStudio のバージョンを確認するには、メニューバーの Help を選択し About RStudio をクリックします（左側）。すると、RStudio のバージョン情報を得ることができます。この場合 2022.07.1 Build 554 となっていることがわかります（右側）。バージョン情報を確認したところで、このパネルを「OK」ボタンをクリックすると閉じられます。

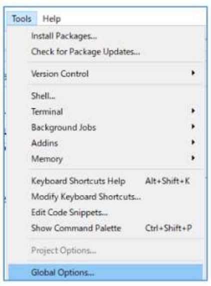


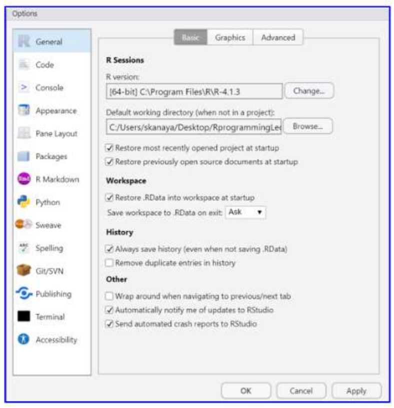


R のバージョンを確認する

続いて R のバージョンを確認しましょう。メニューバーの Tools から Global Options… を選択し、クリックします。すると左側の General として、Rsessions の中の R version:により確認できます。ここでは、[64-bit] C:¥Program Files¥R-4.1.3 となっており、C:¥Program Files¥は R-4.1.3 のイントールフォルダーとなっていることがわかります。また、その下の Default working directory (when not in a project)は、ワーキングデ

ィレクトリフォルダーが示されています。すなわち、データやプログラムの保存先は、このフォルダーとなります。いくつかのフォルダーで設定し、プロジェクトごとにフォルダーを変える時には、Default working directory (when not in a project)の Browse…ボタンをクリックし、選ぶこともできます。ただし、選んだあとに最下段の Apply ボタンをクリックしたのち、RStudio を再起動しなければなりません。これは、メニューバーの一番右側の「x」をクリックすると、RStudio をシャットダウンできます。その後 RStudio を再度立ち上げれば、Default working directory (when not in a project)が更新されていることがわかるでしょう。なお、Options パネルの OK をクリックするとこのパネルを消去することができます。



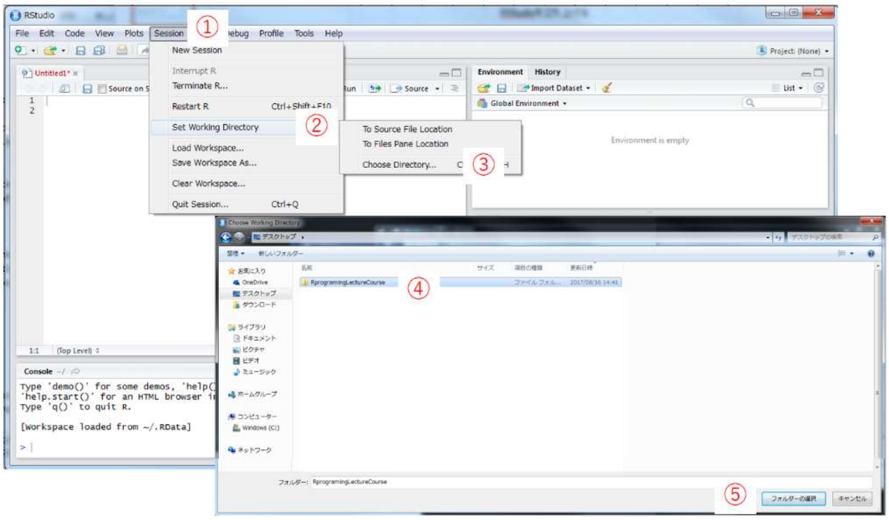


RStudio でプログラムをつくる

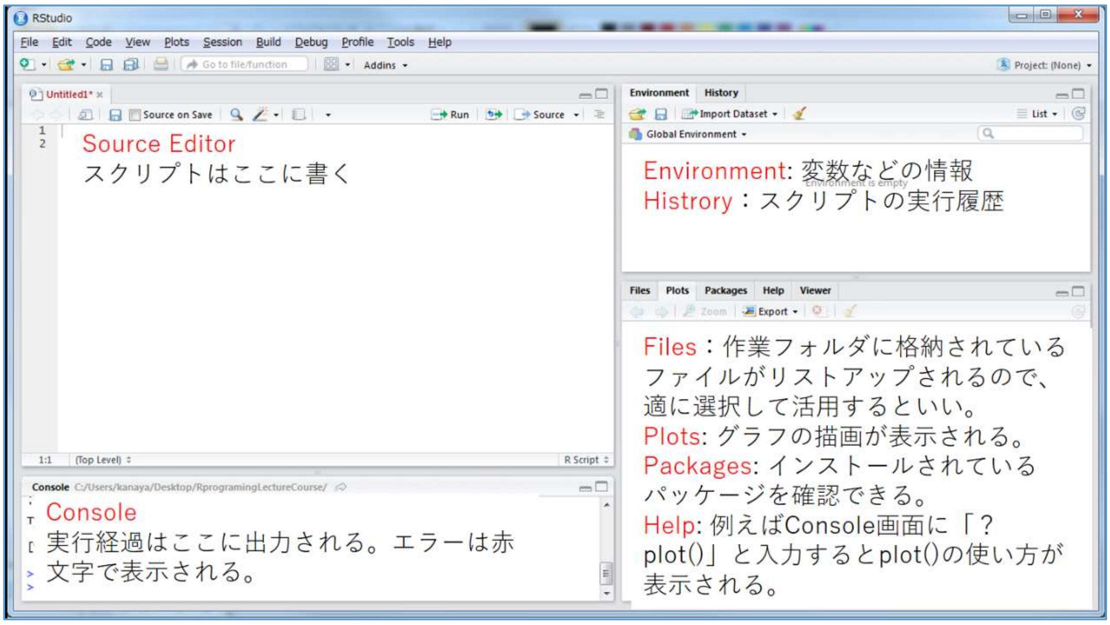
デスクトップに RprogrammingLectureCourse というフォルダーを作りましょう。その後、R を立ち上げます。

続いて、下図に従って、1.メニューから Session、2.Set Working Directory、3.Choose Directory を選びます。

そこで、4.ワーキング・ディレクトリ RprogrammingLectureCourse を下側の画面から選びます。すると、フォルダー名に RprogrammingLectureCourse が入力されるので、5.「フォルダーを選択」ボタンをクリックします。これで、プログラミングの準備が終わりました。



RStudio の構造をみてみましょう。プログラムは Source Editor(左上)に作成します。Source Editor で作成されたプログラムを実行すると Console 画面に出力されます。ここで、プログラムのそれぞれのインストラクション(1 つのプログラムコード)が実行された結果が Console 画面に出力されます。



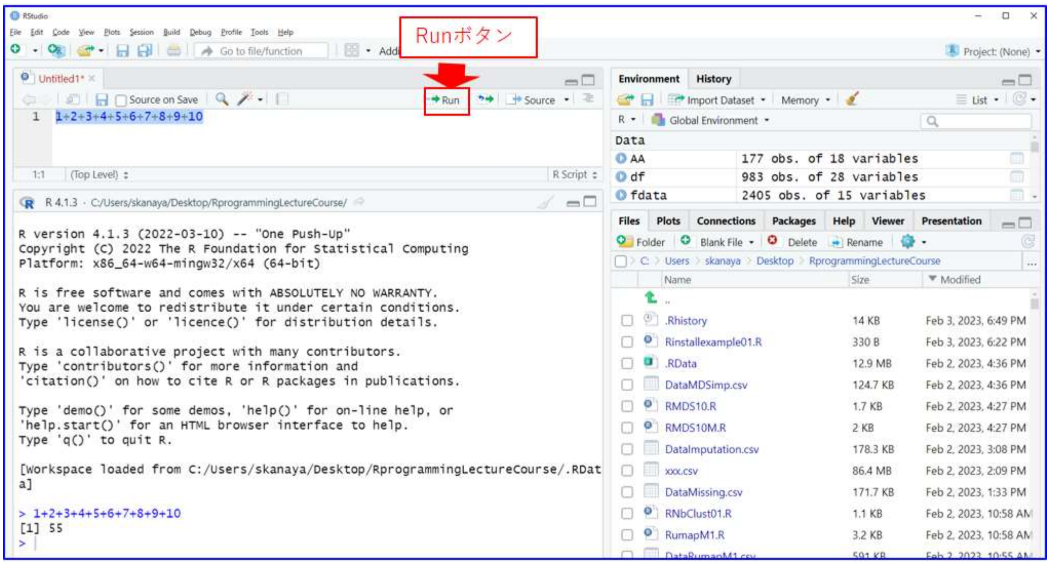
では実際にプログラムを作成してみましょう。まず Source Editor に

「1+2+3+4+5+6+7+8+9+10」

と入力します。この 1 行を選択し、上部の Run ボタンをクリックさせると、プログラムが実行されます(以下の図参照)その結果、コンソールには

> 1+2+3+4+5+6+7+8+9+10
[1] 55

と表示されます。これは、1+2+3+4+5+6+7+8+9+10 を実行した結果が 55 として計算されたことを示しています。



二次方程式$ax^2+bx+c=0$ の解は、$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$ です。これをもとに、二次方程式を解くプログラムを作成してみよう。ここで、二次方程式について、a, b, c が分っているときの x を求めるプログラムは以下のようになります。

```
1  a <- 2
2  b <- 7
3  c <- 5
4  x1 <- (-b + sqrt(b^2 -4 * a * c))/(2 * a)
5  x2 <- (-b - sqrt(b^2 - 4 * a * c)) / (2 * a)
6  x1
7  x2
```

《 プログラムの概要 》

1 行目:a<-1 とは、a に 1 を代入するという操作です。

1-3 行目:a=1, b=7, c=5 が代入されました。

4 行目:$x=\frac{-b+\sqrt{b^2-4ac}}{2a}$

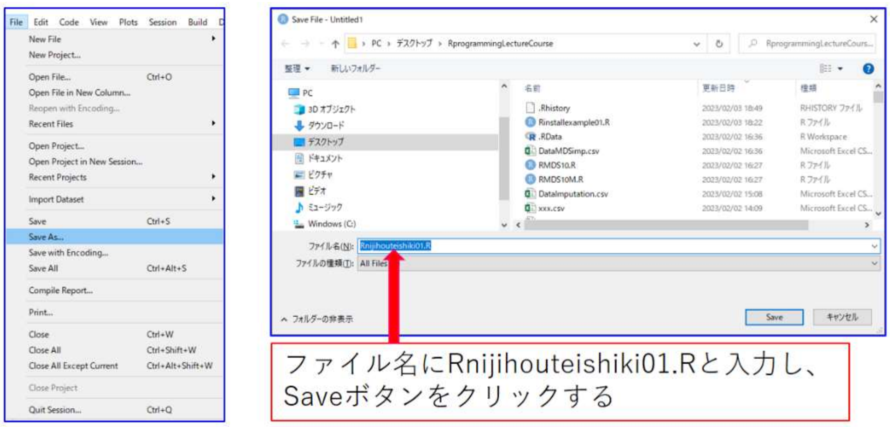
5 行目:$x=\frac{-b-\sqrt{b^2-4ac}}{2a}$

を R により作成しました。ここで sqrt()は√ です。
6,7 行目:x1 と x2 に格納された値(二次方程式の解が出力されます。以下に実行結果を示しました。

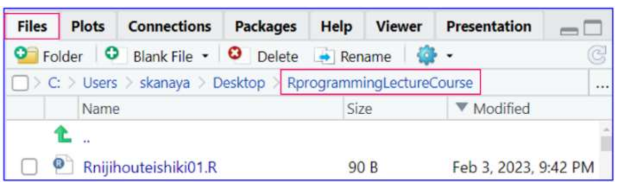
```
> a <- 2
> b <- 7
> c <- 5
> x1 <- (-b + sqrt(b^2 - 4 * a * c))/(2 * a)
> x2 <- (-b - sqrt(b^2   4 * a * c))/(2 * a)
> x1
[1] -1
> x2
[1] -2.5
```

その結果、二次方程式の解が x1 と x2 に格納され、x1 と x2 を出力すると-1 と-2.5 となりました。

ではこのプログラムをファイルにセーブしてみましょう。メニューバーの「Files」→「Save as」を選択し、ファイル名に RNijihouteishiki01.R と入力の後、Save ボタンをクリックします。これで RprogrammingLectureCourse に RNijihouteishiki01.R はセーブされました。



つづいて、右下側のパネルの Files 選択し(左上)、RprogrammingLectureCourse をクリックするとこのフォルダーが更新されます。その結果、Rnijihouteishiki01.R が出現します。これで、ファイル Rnijihouteishiki01.R が RprogrammingLectureCourse フォルダに格納されていることがわかります。



このRnijihouteishiki01.Rファイルをクリックすると、SourceEditor画面（左上）にこのプログラムをロードできます。このようにして、「プログラムを作成・実行し、ワーキング（作業）ディレクトリに保存する。さらにワーキングディレクトリから R プログラムをロードし、さらにプログラムを開発する、あるいは実行し、フォルダにセーブする。」という流れが理解できたと思います。このようにして、これからデータ解析を R プログラムにより解析していきましょう。

著者
金谷 重彦　奈良先端科学技術大学院大学・データ駆動型サイエンス創造センター
平尾 俊貴　奈良先端科学技術大学院大学・先端科学技術研究科・情報科学領域
小笠原 司　奈良先端科学技術大学院大学・地域共創推進室
松本 健一　奈良先端科学技術大学院大学・先端科学技術研究科・情報科学領域

高校生のデータサイエンス・77 本ノック
NAIST STELLA プログラム“「共創」が育む主体性の未来”学習教材

2024 年 3 月 21 日　初版第一刷発行
編著者　金谷重彦
著作　奈良先端科学技術大学院大学
NAIST STELLA プログラム運営委員会
発行所　国立大学法人 奈良先端科学技術大学院大学
〒630-0192 奈良県生駒市高山町 8916-5
https://www.naist.jp/
電話 0743-72-5111（代表）

ISBN978-4-902874-04-4