5. 得点

データ分析の目的によってデータ行列の成分全体を一定の規則で変換することがあります。この節では, データを構成する個々のデータの「得点」(Score)に着目し，データ内のそれぞれの値の特徴をデータ全体の中で観察します。以下で扱う得点の中には「度数」という用語を使って「相対度数」「期待度数」のように一般によく使われるものもありますが，「加重得点」「限定得点」「代表得点」「卓立得点」は一般に使われていません。「標準得点」は「標準スコア」「標準測度」などと呼ばれますが，ここではデータの個々の数値を変換した値を成分としてもつ行列をすべて「得点行列」という用語で統一しました。実測値の分布を見るために使う「階級得点」は，もとの実測値の行列と大きさ(行数と列数)が異なりますが，その他はすべて実測値の大きさと同じです。

次のデータ(D)を統一して使用します。

```
D=c(10,19,14,7,12,11,7,10,0,1,0,0,1,12,1,0,1,2,3,3)
D=matrix(D,4,5,T); rownames(D)='r'&1:4; colnames(D)='c'&1:5
D
   c1 c2 c3 c4 c5
r1 10 19 14  7 12
r2 11  7 10  0  1
r3  0  0  1 12  1
r4  0  1  2  3  3
```

次の関数を随時使用します。

ユーザー関数:

```
"&"=function(x,y){
  if(is.character(c(x, y))) paste0(x, y) else base::"&"(x, y)
} #文字を結合
```

```
Aa=function(D,f=sum,dx=3,dy=3){
  X=Round(as.matrix(D),dx); s=deparse(substitute(f)) #  変数=>文字列
  C=matrix('-',1,ncol(X)); rownames(C)=s; X=rbind(X, C)
  C=matrix('-',nrow(X),1); colnames(C)=s; X=cbind(X, C)
  X[s,s]=Round(f(D),dy); JustR(X)
} #Add all  行列に計算値(全:1)を追加  dx, dy: D, 計算値の小数桁数
```

```
Aa2=function(D,f1=min,f2=max,dx=3,d1=3,d2=3){
  X=Round(as.matrix(D),dx)
  s1=deparse(substitute(f1)) #  変数=>文字列
  s2=deparse(substitute(f2)) #  変数=>文字列
```

```
  C=matrix('-',1,ncol(X)); rownames(C)=s1; X=rbind(X,C)
  C=matrix('-',1,ncol(X)); rownames(C)=s2; X=rbind(X,C)
  C=matrix('-',nrow(X),1); colnames(C)=s1; X=cbind(X,C)
  C=matrix('-',nrow(X),1); colnames(C)=s2; X=cbind(X,C)
  X[s1,s1]=Round(f1(D),d1); X[s2,s2]=Round(f2(D),d2); JustR(X)
} #Add all 行列に計算値(全:2)を追加 dx, d1,d2: D, 計算値の小数桁数
```

```
Ab=function(X,func=mean,dx=3,dy=3){ #d: 小数桁 <Rc
  str=deparse(substitute(func)) # 変数=>文字列
  X1=Ap(X,1,func);          X2=Round(X1,dy);          colnames(X2)=str;
X3=cbind(Round(X,dx),X2)
  Y=Ap(X,2,func); Y1=Round(Y,dy); W=Round(func(X),dy); Y2=t(c(Y1,W))
  rownames(Y2)=str; JustR(rbind(X3,Y2))
} #Add both 行列に計算値(両:1)を追加 dx, dy: D, 計算値の小数桁数
```

```
Ab2=function(D,f1=min,f2=max){ # <Rc2
  s1=deparse(substitute(f1)); s2=deparse(substitute(f2)) #変数=>文字列
  C1=Ap(D,1,f1); colnames(C1)=s1; C2=Ap(D,1,f2); colnames(C2)=s2
  R1=Ap(D,2,f1); rownames(R1)=s1; R2=Ap(D,2,f2); rownames(R2)=s2
  X=cbind(D,C1,C2); Y=rbind(R1,R2)
  Z=BindV(X,Y); Z[s1,s1]=f1(D); Z[s2,s2]=f2(D); JustR(Z)
} ##Add both 行列に計算値(両:2)を追加 dx, d1,d2: D, 計算値の小数桁数
```

```
Ac=function(X,func=mean,dx=3,dy=3){ #dx, dy: 小数桁 <Rv
  str=deparse(substitute(func)) # 変数=>文字列
  X1=Ap(X,2,func);          X2=Round(X1,dy);          rownames(X2)=str;
X3=rbind(Round(X,dx),X2)
  JustR(X3)
} #Add column 行列に計算値(列:1)を追加 dx, dy: D, 計算値の小数桁数
```

```
Ac2=function(X,f1=mean,f2=median,dx=3,d1=3,d2=3){ #dx, d1, d2: 小数桁 <Rv2
  s1=deparse(substitute(f1)) # 変数=>文字列
  s2=deparse(substitute(f2)) # 変数=>文字列
  Z1a=Ap(X,2,f1); Z1b=Round(Z1a,d1); rownames(Z1b)=s1
  Z2a=Ap(X,2,f2); Z2b=Round(Z2a,d2); rownames(Z2b)=s2
  Z=rbind(Round(X,dx),Z1b,Z2b); JustR(Z)
} ##Add column 行列に計算値(列:2)を追加 dx, d1,d2: D, 計算値の小数桁数
```

```
Ar=function(X,func=mean,dx=3,dy=3){ #dx, dy: 小数桁 <Rh
```

```
  str=deparse(substitute(func)) # 変数=>文字列
  X1=Ap(X,1,func);          X2=Round(X1,dy);          colnames(X2)=str;
X3=cbind(Round(X,dx),X2)
  JustR(X3)
} #Add row 行列に計算値(行:1)を追加 dx, dy: D, 計算値の小数桁数
```

```
Ar2=function(X,f1=mean,f2=median,dx=3,d1=3,d2=3){ #dx, d1, d2: 小数桁  <Rh2
  s1=deparse(substitute(f1)) # 変数=>文字列
  s2=deparse(substitute(f2)) # 変数=>文字列
  Z1a=Ap(X,1,f1); Z1b=Round(Z1a,d1); colnames(Z1b)=s1
  Z2a=Ap(X,1,f2); Z2b=Round(Z2a,d2); colnames(Z2b)=s2
  Z=cbind(Round(X,dx),Z1b,Z2b); JustR(Z)
} ##Add row 行列に計算値(行:2)を追加 dx, d1,d2: D, 計算値の小数桁数
```

```
Ap=function(D,s=1,f=sum){
  str=deparse(substitute(f)) # 変数=>文字列
  if(s==1) {W=as.matrix(apply(D,s,f)); colnames(W)=str; W} #縦ベクトル
  else if(s==2) {W=t(apply(D,s,f)); rownames(W)=str; W} #横ベクトル
  else f(D) #スカラー
} # 汎用 apply ex: Ap(D,2,sum)
```

```
Bind=function(X,Y,f=''){
  X=as.matrix(X); Y=as.matrix(Y); rx=nrow(X); ry=nrow(Y)
  if(rx>ry) Y=rbind(Y, matrix(f,rx-ry,ncol(Y)))
  if(rx<ry) X=rbind(X, matrix(f,ry-rx,ncol(X)))
  W=cbind(X,lxx=rep(':',nrow(X)),Y)
  colnames(W)=gsub('lxx',':',colnames(W)); JustR(W)
} #行列を横結合 X:Y (行数不一致可) f: filler
```

```
BIND=function(..., f=''){
  L=list(...); W=L[[1]]; for(i in 2:length(L)) W=Bind(W,L[[i]],f); W
} # 行列を横連結 BIND(A,B,C) =>A:B:C
```

```
BindV=function(X,Y,f=''){
  cx=ncol(X); cy=ncol(Y)
  if(cx>cy) {Y=cbind(Y,matrix(rep(f,nrow(Y)*(cx-cy)),nrow(Y))); C=colnames(X)}
  if(cx<cy) {X=cbind(X,matrix(rep(f,nrow(X)*(cy-cx)),nrow(X))); C=colnames(Y)}
  Z=JustR(rbind(X,Y)); colnames(Z)=C; Z
} # 列数の異なる行列を縦に連結 f: filler
```

```
JustR=function(X){
  for(i in 1:ncol(X)){
    X[,i]=as.character(X[,i]); X[is.na(X)]='NA'
    m=max(nchar(X[,i]),nchar(colnames(X)[i]))
    X[,i]=sprintf('%'&m&'s', X[,i])
    colnames(X)[i]=sprintf('%'&m&'s', colnames(X)[i])
  }; noquote(X)
} #小数数値行列を右揃え
```

```
R=function(X,d=3){ #X: データ, d:小数点以下桁数
  W=X; for(i in 1:ncol(X)){Z=W[,i]
  Z=format(round(X[,i],d),nsmall=d)
  if(all(X<1)) Z=gsub('^0¥¥.','.',Z)
  m=max(nchar(Z)); l=nchar(colnames(W)[i]); W[,i]=Z
  colnames(W)[i]=paste(rep(' ',max(m-l,0)),collapse="")&colnames(W)[i]
  }; noquote(W)
} #小数点以下桁数で丸める (行列)
```

5.1. 相対得点

はじめに行和(横和)と列和(縦和)を計算して，D に連結します。

```
Ab(D,sum,0,0) #データ＋和
     c1 c2 c3 c4 c5 sum
r1   10 19 14  7 12  62
r2   11  7 10  0  1  29
r3    0  0  1 12  1  14
r4    0  1  2  3  3   9
sum 21 27 27 22 17 114
```

この実測値の行列 D では，たとえば r1:c1 の 10 と r2:c1 の 11 をそのまま比較することができません。それぞれの横和(62, 29)が異なるからです。そこで有効になるのが「相対得点」(Relative Score: RS)です。

「行相対得点」(Relative Score in row: RSr)と「列相対得点」(Relative Score in column: RSc)をそれぞの行和(Rs)と列和(Cs)を使って計算します。

```
X=Ar(D,sum,0,0); Y=Ar(RS(D,1),sum,3,3); Bind(X,Y) #データ：相対得点(行)
   c1 c2 c3 c4 c5 sum :    c1    c2    c3    c4    c5   sum
r1 10 19 14  7 12  62 :  .161  .306  .226  .113  .194 1.000
r2 11  7 10  0  1  29 :  .379  .241  .345  .000  .034 1.000
r3  0  0  1 12  1  14 :  .000  .000  .071  .857  .071 1.000
r4  0  1  2  3  3   9 :  .000  .111  .222  .333  .333 1.000

X=Ac(D,sum,0,0); Y=Ac(RS(D,2),sum,3,3); Bind(X,Y) #データ：相対得点(列)
    c1 c2 c3 c4 c5 :    c1    c2    c3    c4    c5
r1  10 19 14  7 12 :  .476  .704  .519  .318  .706
r2  11  7 10  0  1 :  .524  .259  .370  .000  .059
r3   0  0  1 12  1 :  .000  .000  .037  .545  .059
r4   0  1  2  3  3 :  .000  .037  .074  .136  .176
sum 21 27 27 22 17 : 1.000 1.000 1.000 1.000 1.000
```

「全相対得点」(Relative Score in all: RSa)は，それぞれのセルの値を全範囲の和(S)（スカラー）で割ったものです。

```
X=Aa(D,sum,0,0); Y=Aa(RS(D,3),sum,3,3); Bind(X,Y) #データ：相対得点(全)
    c1 c2 c3 c4 c5 sum :    c1    c2    c3    c4    c5   sum
r1  10 19 14  7 12   - :  .088  .167  .123  .061  .105     -
r2  11  7 10  0  1   - :  .096  .061  .088  .000  .009     -
r3   0  0  1 12  1   - :  .000  .000  .009  .105  .009     -
r4   0  1  2  3  3   - :  .000  .009  .018  .026  .026     -
sum  -  -  -  -  - 114 :     -     -     -     -     - 1.000
```

「全相対得点」は「総和による標準化」(Normalization by sum: Ns)を示します。単純にデータ X_{np} のそれぞれの成分をデータ総和(T)で割った値です。たとえば，d1:v1 のセルでは 10 / 114 = .088 となります。

ユーザー関数:

```
RS=function(X,s=1) MV(X,Ap(X,s,sum)) #相対得点(s=1:行,2:列,3:全)
```

＊参照：池田(1976: 121-123)

●パーセント・パーミル

次のように相対頻度に 100 を掛けるとパーセントになります。

```
X=Ar(RS(D,1),sum,3,3); Y=Ar(RS(D,1)*100,sum,1,1); Bind(X,Y) #パーセント：行
     c1    c2    c3    c4    c5   sum :   c1   c2   c3   c4   c5   sum
r1 .161  .306  .226  .113  .194 1.000 : 16.1 30.6 22.6 11.3 19.4 100.0
r2 .379  .241  .345  .000  .034 1.000 : 37.9 24.1 34.5  0.0  3.4 100.0
r3 .000  .000  .071  .857  .071 1.000 :  0.0  0.0  7.1 85.7  7.1 100.0
r4 .000  .111  .222  .333  .333 1.000 :  0.0 11.1 22.2 33.3 33.3 100.0
```

次のように乗数として 1000 を使うとパーミルになります。

```
X=Ar(RS(D,1),sum,3,3); Y=Ar(RS(D,1)*1000,sum,1,1); Bind(X,Y) #パーミル: 行
      c1    c2    c3    c4    c5   sum :    c1    c2    c3    c4    c5    sum
r1  .161  .306  .226  .113  .194 1.000 : 161.3 306.5 225.8 112.9 193.5 1000.0
r2  .379  .241  .345  .000  .034 1.000 : 379.3 241.4 344.8   0.0  34.5 1000.0
r3  .000  .000  .071  .857  .071 1.000 :   0.0   0.0  71.4 857.1  71.4 1000.0
r4  .000  .111  .222  .333  .333 1.000 :   0.0 111.1 222.2 333.3 333.3 1000.0
```

5.2. 正規化得点

相対頻度のパーミルでは乗数を 1000 としましたが, コーパス言語学では 10^6 (=1000000)を乗数とするパーミリオンがよく使われます。そのとき, 相対頻度のように母数として表の中の和ではなく, 次のような総語数(W)が使われることがあります。

```
W    c1     c2      c3      c4     c5
& 62523 293945 1149932 4404011 632944
```

次は, データ(D)を:の左に示し, データ(D)を W で割って 10^6 を掛けたパーミリオンを:の右に示したものです。

```
Bind(D,R(NS(D,W,10^6),1))
   c1 c2 c3 c4 c5      c1   c2   c3  c4   c5
r1 10 19 14  7 12 : 159.9 64.6 12.2 1.6 19.0
r2 11  7 10  0  1 : 175.9 23.8  8.7 0.0  1.6
r3  0  0  1 12  1 :   0.0  0.0  0.9 2.7  1.6
r4  0  1  2  3  3 :   0.0  3.4  1.7 0.7  4.7
```

乗数は 10^6 以外にも 10^3, 10^4, 10^5 などを使うこともできます。乗数の選択は母数の最小値の桁数を使えば, 過度の外挿(extrapolation: 既知の数値データを基にして、そのデータの範囲の外側で予想される数値を求めること)を防ぐことができます。上の例では最小値は 62523 なので, その桁数 5 から 1 を引いた数を使って乗数を 10^4 (=10000)とします[1]。

> NSm(W) # 10000

```
Bind(D,R(NS(D,W,10^4), 3))
   c1 c2 c3 c4 c5 :    c1    c2    c3    c4    c5
r1 10 19 14  7 12 : 1.599  .646  .122  .016  .190
r2 11  7 10  0  1 : 1.759  .238  .087  .000  .016
r3  0  0  1 12  1 :  .000  .000  .009  .027  .016
r4  0  1  2  3  3 :  .000  .034  .017  .007  .047
```

ユーザー関数:

```
NS=function(X,W,m=0){
```

[1] 出力される数値の桁数(出力桁数)を 4 とするために, 小数桁は 3 としました。小数桁を 1 とすると, 出力桁数が 2 となって比較が困難になります。

```
  if(m==0) m=10^(nchar(as.character(min(W)))-1); MV(D,W)*m
} # 正規化得点(X: データ行列, W: 母数ベクトル, m:乗数)
```

ユーザー関数:

```
NSm=function(W){10^(nchar(as.character(min(W)))-1)} # 正規化得点乗数
```

■2 重子音を含む語

次は中世スペイン語(公証文書)の鼻音と流音の 2 重子音文字を有する語の頻度の実測値(X)です。

```
X   1260 1280 1300 1320 1340 1360 1380 1400 1420 1440 1460 1480 1500
nn   550   66  143   57    1    2    2    4    4    1    0    2   30
ll  2310 1166 4524 1354  243  367  325  571  902  217  439  589  776
rr   625  327 1563  846  109  309  283  533  290  181  152  249  273
```

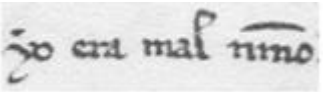
次は年代ごとのすべての文書の総語数(W)を示します。

```
W   1260  1280   1300  1320 1340  1360  1380  1400  1420 1440  1460  1480  1500
& 62549 29396 114499 44040 6000 11732 10506 19276 27990 8131 15952 20792 27048
```

次はデータ(X)と文書の総語数(W)を使った正規化頻度を示します。このとき，乗数は総語数の最小値 6000 に合わせて 1000 にしました。

```
min(W); NSm(W); R(NS(X,W),1)
[1] 6000
[1] 1000
   1260 1280 1300 1320 1340 1360 1380 1400 1420 1440 1460 1480 1500
nn  8.8  2.2  1.2  1.3  0.2  0.2  0.2  0.2  0.1  0.1  0.0  0.1  1.1
ll 36.9 39.7 39.5 30.7 40.5 31.3 30.9 29.6 32.2 26.7 27.5 28.3 28.7
rr 10.0 11.1 13.7 19.2 18.2 26.3 26.9 27.7 10.4 22.3  9.5 12.0 10.1
```

14 世紀に nn が急減したのは，これが n の上に省略記号の⁀を付けた形に変わったためです。これがスペイン語特有の文字エニェ(ñ)の起源になりました。



yo era mas ni<n>no「私は幼少だった」

5.3. 標準化得点

下のようなデータ行列(1)のそれぞれのセルを一定の値で割って，縦和も横和もそれぞれすべての和が同一になるように変換すると，全体の中で値を相対的に見ることができます。

```
X=Ab(D,sum,0,0); Y=Ab(Mosteller(D),sum,3,3); Bind(X,Y)
#モステラーの標準化
    c1 c2 c3 c4 c5 sum :    c1    c2    c3    c4    c5   sum
r1  10 19 14  7 12  62 :  .068  .091  .043  .007  .041  .250
r2  11  7 10  0  1  29 :  .132  .059  .053  .000  .006  .250
r3   0  0  1 12  1  14 :  .000  .000  .041  .162  .047  .250
r4   0  1  2  3  3   9 :  .000  .050  .063  .031  .106  .250
sum 21 27 27 22 17 114 :  .200  .200  .200  .200  .200 1.000
```

手順(1)：はじめに，行和 Rs を一定に揃え，総和を 1 にするために，行列全体を行和*行数(=4)で割ります。

```
X=MV(D,Rsums(D),'d')/4; Ab(X,sum,3,3) #Step-1
       c1    c2    c3    c4    c5   sum
r1   .040  .077  .056  .028  .048  .250
r2   .095  .060  .086  .000  .009  .250
r3   .000  .000  .018  .214  .018  .250
r4   .000  .028  .056  .083  .083  .250
sum  .135  .165  .216  .326  .158 1.000
```

手順(2)：次に，列和 Cs を一定に揃え総和を 1 にするためは，行列全体を列和 Sv*列数(=5)で割ります。

```
X=MV(X,Csums(X),'d')/5; Ab(X,sum,3,3) #Step-2
       c1    c2    c3    c4    c5   sum
r1   .060  .093  .052  .017  .061  .283
r2   .140  .073  .080  .000  .011  .304
r3   .000  .000  .017  .132  .023  .171
r4   .000  .034  .051  .051  .105  .242
sum  .200  .200  .200  .200  .200 1.000
```

このとき行和 Rs が変化しますから，再び行和 Cs*行数で割り，次に列和 Sv*列数で割る，という演算をします。この演算をセルの値の変化がほとんでなくなるまで繰り返します。その結果，先の(2)のようになります。

ユーザー関数:

```
Mosteller=function(D){
  for(i in 1:1000){
    X=D; D=MV(D,Rsums(D),'d')/nrow(D); D=MV(D,Csums(D),'d')/ncol(D)
    #X=D; D=D/rowSums(D)/nrow(D); D=t(t(D)/colSums(D))/ncol(D)
    if(sum((D-X)^2)<10^-10) break
  }; D
}# モステラーの標準化 (Mosteller standardization)
```

この方法は「モステラーの標準化」(Mosteller standardization)と呼ばれます(＊池田 1976: 123-124)。モステラーの標準化はすべての行和を同じに揃え，すべての列和を同じに揃えます。そのため個々のデータの規模が元の

数値と大きく異なることになるので(たとえば上の出力でマークした数値), 注意が必要です。

■ «s»の 2 異文字・死亡者と生存者

中世・近代スペイン語では，文字«s»に，短い<s> と縦長の<ſ>という異文字(allograph)がありました。その分布の特徴は語末に短い<s> が使われる傾向があった，ということが観察されています。しかし，確かに語末で短い<s>が多く使われているのですが，それは語頭・語中でもやはり使われています。一方，縦長の<ſ>は語頭・語中で多く使われていますが，語末でもかなり見つかります(下左表：『アレクサンダー大王物語』 *Libro de Alexandre* (1300)の冒頭から 2 万字まで)。

*	Initial	Medial	Final
<s>	62	2	593
<ſ>	314	412	109

短い<s>が語末に出現する傾向は，このような小さな規模（2 行 3 列の大きさの表）で頻度が低いデータ（総数 1492）ならばとくに標準化しなくても大体様子がわかるのですが，データの規模とスケールがさらに大きくなると，分布の傾向を見ることが難しくなります。そこで，よく行われるのは次のような横和，または縦和で割った相対頻度の表示です。

```
X=Ab(W,sum,0,0); Y=Ab(RS(W,1),sum,3,3); Bind(X,Y) #相対得点：行
    Initial Medial Final  sum : Initial Medial Final   sum
<s>      62      2   593  657 :    .094   .003  .903 1.000
<ſ>     314    412   109  835 :   .376   .493  .131 1.000
sum     376    414   702 1492 :    .470   .496 1.033 2.000

X=Ab(W,sum,0,0); Y=Ab(RS(W,2),sum,3,3); Bind(X,Y) #相対得点：列
    Initial Medial Final  sum : Initial Medial Final   sum
<s>      62      2   593  657 :    .165   .005  .845 1.014
<ſ>     314    412   109  835 :   .835   .995  .155 1.986
sum     376    414   702 1492 :   1.000  1.000 1.000 3.000
```

ここで，横和(Rs)で相対化すると(相対得点：行)，観点は横軸に集中し，縦長の<ſ>が語頭・語中でやや多いようですが，どちらも半数を超えていません(.376, .493)。一方，縦和で相対化すると(相対得点：列)，非常に高い比率(.835, .995)を占めています。行相対得点ではそれぞれの文字の出現位置を調べます。列相対得点ではそれぞれの位置文字の出現位置を調べます。

下右表は「総和による標準化」(全相対得点)の結果です。全体で見るならば，注目すべきは語末の<s>になります。

```
Bind(R(RCsum(X),0), R(RCsum(RS(X,3)))) #全相対得点
    Initial Medial Final   Rs : Initial Medial Final    Rs
<s>      62      2   593  657 :    .042   .001  .397  .440
<ſ>     314    412   109  835 :    .210   .276  .073  .560
Cs      376    414   702 1492 :    .252   .277  .471 1.000
```

このように，同じデータを扱いながら，相対頻度では視点によって(行・列・全体)観察が異なってきます。そして，次の「モステラーの標準化」の結果は「総和による標準化」と総和は同じですが，それぞれのセルの値は大きく異なります。

```
Bind(R(RCsum(X),0), R(RCsum(Mosteller(X)))) #モステラーの標準化
    Initial Medial Final   Rs : Initial Medial Final    Rs
<s>      62      2   593  657 :    .170   .008  .322  .500
<ſ>     314    412   109  835 :    .164   .325  .011  .500
Cs      376    414   702 1492 :    .333   .333  .333 1.000
```

総和による標準化はデータ行列全体を総和で割ったものなので，データ行列そのものを観察していることと大きな違いはありません。総和が 1 になるので，全体の中での比率を観察していることになります。モステラーの標準化は行和と列和がそれぞれ同じになるので，それぞれを同じ規模にして比較することができます。しかし，データの比率は保持されていません。たとえばデータの最初のセル 62 を行和 657 で割ると 62/657=.094，列和 366 で割ると 62/376=.165，総和で割ると，62/1492=.042 になっています。また，Initial の列の 2 つ数値，.170 と.164 の大小関係と差は，データの 62 と 314 と大小関係と差が大きく異なっています。

5.4. 卓立得点

[1] 行卓立相対得点と列卓立相対得点

「自分（セル）が他のメンバー（セル）たちと違う」ことを示す「卓立得点」(Prominent Score: PS)という数値を提案します。ここでは 1 つのセルの値(X)，たとえば D[1,1]=D['r1','c1']=10 を取り出して説明します。

```
Ab(D,sum,0,0) #データ＋和
    c1 c2 c3 c4 c5 sum
r1  10 19 14  7 12  62
r2  11  7 10  0  1  29
r3   0  0  1 12  1  14
r4   0  1  2  3  3   9
sum 21 27 27 22 17 114
```

ここで，x=D[1,1]の実測値(=10)を，横行の他の値全体の和(Rs - x = 62 - 10 = 52)と比較します。このとき，そのまま比較するのではなく，X に列数 p -

1 = 5 - 1 = 4 を掛けた値(p-1)*x と Rs-x を比較します。これは x (1 個)の大きさを他のセル全部(p-1)個と比べられないからです。そこで，セルの数を同じと見なしたときの x の値，(p-1)*x を考えます。(p-1)*x を，他の Rs-x と相対化した値は(p-1)*x / [(p-1)*x + (Rs-x)]です。これを「行卓立相対得点」(Prominent Relative Score in Row: PRSR)とします。卓立係数(PS)は[0.0 ~ 1.0]の範囲になります。

$$\begin{aligned} PRSR &= (p-1)*D / [(p-1)*D + (Rs-D)] \\ &= (p-1)* D / [(p-2)*D + Rs] \end{aligned}$$

同様に「列卓立相対得点」(Prominent Relative Score in Column: PRSC)は

$$\begin{aligned} PRSC &= (n-1)*D / [(n-1)*D + (Cs-D)] \\ &= (n-1)*D / [(n-2)*D + Cs] \end{aligned}$$

セルの数が多くなると，相対得点は小さくなりがちですが，卓立相対得点(PRS)ではセルの数(Cn)の大小にあまり左右されない数値が得られます。これは PS の式の分子にも分母にも p, n があるためです。

```
X=Ar(D,sum,0,0); Y=Ar(PRS(D,1),sum,3,3); Bind(X,Y) #データ：卓立得点：行
   c1 c2 c3 c4 c5 sum :    c1    c2    c3    c4    c5   sum
r1 10 19 14  7 12  62 :  .435  .639  .538  .337  .490 2.439
r2 11  7 10  0  1  29 :  .710  .560  .678  .000  .125 2.073
r3  0  0  1 12  1  14 :  .000  .000  .235  .960  .235 1.431
r4  0  1  2  3  3   9 :  .000  .333  .533  .667  .667 2.200

X=Ac(D,sum,0,0); Y=Ac(PRS(D,2),sum,3,3); Bind(X,Y) #データ：卓立得点：列
    c1 c2 c3 c4 c5 :    c1    c2    c3    c4    c5
r1  10 19 14  7 12 :  .732  .877  .764  .583  .878
r2  11  7 10  0  1 :  .767  .512  .638  .000  .158
r3   0  0  1 12  1 :  .000  .000  .103  .783  .158
r4   0  1  2  3  3 :  .000  .103  .194  .321  .391
sum 21 27 27 22 17 : 1.499 1.493 1.699 1.687 1.585
```

[2] 全卓立相対得点

「全卓立相対得点」(Prominent Relative Score in All: PRSA)は x を行列全体のその他のメンバーの和(S - x) と比較します(S: 総和)。そのとき，x には行列全体の個数(n*p-1)を加重して比べられるようにします。

$$\begin{aligned} PRSA &= (n*p-1)*D / [(n*p-1)*D + (S - D)] \\ &= (n*p-1)*D / [(n*p-2)*D + S] \end{aligned}$$

```
X=Aa(D,sum,0,0); Y=Aa(PRS(D,3),sum,3,3); Bind(X,Y) #データ：卓立得点： 全
    c1 c2 c3 c4 c5 sum :    c1    c2    c3    c4    c5   sum
r1  10 19 14  7 12   - :  .646  .792  .727  .554  .691     -
r2  11  7 10  0  1   - :  .670  .554  .646  .000  .144     -
r3   0  0  1 12  1   - :  .000  .000  .144  .691  .144     -
r4   0  1  2  3  3   - :  .000  .144  .253  .339  .339     -
sum  -  -  -  -  - 114 :     -     -     -     -     - 7.479
```

ユーザー関数:

```
PRS=function(D,s=1){
  n=nrow(D); p=ncol(D); Rs=rowSums(D); Cs=colSums(D)
  if(s==1) (p-1)*D/((p-2)*D+Rs)
  else if(s==2) (n-1)*D/t(t((n-2)*D)+Cs)
  else (n*p-1)*D/((n*p-2)*D+sum(D))
} # 卓立相対得点(Prominent Relative Score, s=1: 行, 2:列, 3:全)
```

■中世・近代スペイン公証文書の略記形

中世・近代スペインの公証文書では, 多くの語が完全な形ではなく語中・語尾が省略されて書かれていました。次の表は頻繁に使われた略記形の頻度（千語率(パーミル)を分数平均で両軸相対化：→「得点」「相対得点」）を示します。

NS.FM	d<e>	d<e>l	d<e>la	d<e>los	d<ic>ha	d<ic>ho	d<ic>hos	dich<o>
1260	348	61	28	174				22
1280	100	66	71					541
1300	629	824	922	686	3			2556
1320	1048	1087	1016	438				5250
1340	215	237	379	103				833
1360	1196	702	805	273				2289
1380	906	1147	1081	451	37	23	65	1372
1400	545	387	396	210	13	24	27	706
1420	981	847	517	331	153	299	195	63
1440	989	1354	938	138	461	548	233	18
1460	914	473	397	158	250	306	303	
1480	2623	1669	902	201	1118	1164	598	
1500	1465	1207	811	412	776	687	541	10
1520	1503	1110	629	231	1021	1052	667	
1540	2707	1854	842	284	1315	1719	865	
1560	660	481	280	121	1533	1192	901	
1580	154	52	88		554	611	457	
1600	558	378			1490	1566	1049	
1620	30				93	63	74	
1640	66				1570	1932	1170	
1660					288	229	78	
1680	43				3566	4953	2579	

上の分布を見ると略記形が年代によって集中していることがわかります。d<e>はどのバリアント(d<e>, d<e>l, d<e>la, d<e>los)も 15 世紀後半を頂点とし, d<ic>ho のタイプの略記形(d<ic>ha, d<ic>ho, d<ic>hos)は 16 世紀前半を頂点としています。一方, dich<o>のように語尾が脱落するのは, かなり早期に見られるので（14 世紀前半）, これは略記というよりも当時頻繁に起きた語末母音の脱落によるものだと思います。それぞれの語形が一定の年代に集中しているので公証人たちが当時の規範に従っていたことを示しています。

5.5. 比例得点

絶対頻度と相対頻度はそれぞれの特徴があるので, データを観察するときに併用されることがあります。一般に頻度を比較するときは相対頻度が使われますが, 相対頻度の計算で分母の規模が大きく異なるとき比較が困難になります。次に, その 1 つの解決法を提案します。

```
Ab(D,sum,0,0) #データ＋和
     c1 c2 c3 c4 c5 sum
r1   10 19 14  7 12  62
r2   11  7 10  0  1  29
r3    0  0  1 12  1  14
r4    0  1  2  3  3   9
sum 21 27 27 22 17 114
```

たとえば，r1:c2 の 19 はその横和が 62 ですから，この相対得点は 19/62 = .306 になります。一方，r4:c4 の 3 の相対得点は 3/9 = .333 になり，r1:c2 よりも大きな値になります。しかし，私たちの直感では前者の 19 のほうが後者の 3 よりも「重い」値だと感じられます。

```
X=Ar(D,sum,0,0); Y=Ar(RS(D,1),sum,3,3); Bind(X,Y) #データ：相対得点：行
   c1 c2 c3 c4 c5 sum :    c1    c2    c3    c4    c5   sum
r1 10 19 14  7 12  62 :  .161  .306  .226  .113  .194 1.000
r2 11  7 10  0  1  29 :  .379  .241  .345  .000  .034 1.000
r3  0  0  1 12  1  14 :  .000  .000  .071  .857  .071 1.000
r4  0  1  2  3  3   9 :  .000  .111  .222  .333  .333 1.000
```

このように実測値の得点を比較するとき，その実測値(OS)と相対得点(RS)の積にすると，実態を表す数値として直感的に納得がいくことがあります。実測値に相対得点という重みを与えたからです。たとえば，上表の r1:c2 の 19 には 19 / 62 = .306 という重みを与え，r4:c2 の 3 には 3 / 9 = .333 という重みを与えます。そこで「**比例得点**」(Proportional Score: PS)として次の式を提案します。

PSR = D*RSR = D*D / Rs = D^2 / Rs
PSR.: 0.0 (D = 0) ≦ 0.5 (D^2 = Rs / 2) ≦ D (D=Rs)

PSC = D*RSC = D*D / Cs = D^2 / Cs
PSC.: 0.0 (D = 0) ≦ 0.5 (D^2 = Cs / 2) ≦ D (D=Cs)

ここで，PSR は**行比例得点**，PSC は**列比例得点**，D は実測値，RSR は行相対得点，Rs は横和列（縦ベクトル），RSC は列相対得点，Cs は縦和行（横ベクトル）を示します。比例得点(PS)は D = 0 のときに最小値ゼロになり，D=Rs, D=Cs のとき，つまりデータの中に D 以外の数値がないときに最大値 D になります。

```
X=Ar(D,sum,0,0); Y=Ar(PS(D,1),sum,1,1); Bind(X,Y) #データ：比例得点：行
   c1 c2 c3 c4 c5 sum :   c1   c2   c3   c4   c5  sum
r1 10 19 14  7 12  62 :  1.6  5.8  3.2  0.8  2.3 13.7
r2 11  7 10  0  1  29 :  4.2  1.7  3.4  0.0  0.0  9.3
r3  0  0  1 12  1  14 :  0.0  0.0  0.1 10.3  0.1 10.4
r4  0  1  2  3  3   9 :  0.0  0.1  0.4  1.0  1.0  2.6
```

```
X=Ac(D,sum,0,0); Y=Ac(PS(D,2),sum,1,1); Bind(X,Y) #データ：比例得点: 列
    c1 c2 c3 c4 c5 :   c1   c2   c3   c4   c5
r1  10 19 14  7 12 :  4.8 13.4  7.3  2.2  8.5
r2  11  7 10  0  1 :  5.8  1.8  3.7  0.0  0.1
r3   0  0  1 12  1 :  0.0  0.0  0.0  6.5  0.1
r4   0  1  2  3  3 :  0.0  0.0  0.1  0.4  0.5
sum 21 27 27 22 17 : 10.5 15.2 11.1  9.2  9.1
```

全比例得点(Proportional Score in All: PSA)を求めるには，分母に全得点の総和(S)を使います。全体比例得点(PSA)は表全体の総和(N)で相対化されるために全体的に数値が低くなる傾向があります。

PSA = D^2 / S

```
X=Aa(D,sum,0,0); Y=Aa(PS(D,3),sum,1,1); Bind(X,Y) #データ：比例得点: 全
    c1 c2 c3 c4 c5 sum :  c1  c2  c3  c4  c5  sum
r1  10 19 14  7 12   - :  .9 3.2 1.7  .4 1.3    -
r2  11  7 10  0  1   - : 1.1  .4  .9  .0  .0    -
r3   0  0  1 12  1   - :  .0  .0  .0 1.3  .0    -
r4   0  1  2  3  3   - :  .0  .0  .0  .1  .1    -
sum  -  -  -  -  - 114 :   -   -   -   -   - 11.3
```

ユーザー関数:

```
PS=function(X,s=1) X*RS(X,s) #比例得点  (s=1:行; 2:列; 3:全)
```

●打率と安打数

たとえば，1 シーズンに 10 打数 3 安打という成績の野球選手 A と 100 打数 25 安打の選手 B の成績を比べるとき，打率だけを見ると 0.3 と 0.25 になるので，A のほうが優秀，ということになります。しかし，安打数で比べるならば後者 B のほうがずっとチームの成績に貢献しています。これを比例得点で比べるならば，0.9 と 6.25 という数値になり，後者 B のほうが前者 A の 7 倍近い成績(6.944)になります。このように数値の評価をするときは，実測値（安打数）や相対得点（打率）よりも比例得点のほうが直感に合う数値になるでしょう。

●絶対頻度・相対得点・比例得点の比較

1/10 と 10/100 を比べると，どちらも相対得点は同じですが(.100, 10%)，比例得点は，それぞれ.100, 1.000 となって，両者の差は.900 になり，後者のほうがかなり「重い」値になります。次に，x / a と y / b の，それぞれの値を次の表で比較しましょう。RS は相対得点，PS は比例得点，D は比例得点の差を示します。y を 10 から 0 まで-1 のステップで下げていきます。

x	a	**y**	b	RS(x)	RS(y)	PS(x)	PS(y)	D=PS(y)-PS(x)
1	10	**10**	100	.100	.100	.100	1.000	**.900**
1	10	**9**	100	.100	.090	.100	.810	**.710**
1	10	**8**	100	.100	.080	.100	.640	**.540**
1	10	**7**	100	.100	.070	.100	.490	**.390**
1	10	**6**	100	.100	.060	.100	.360	**.260**
1	10	**5**	100	.100	.050	.100	.250	**.150**
1	10	**4**	100	.100	.040	.100	.160	**.060**
1	10	**3**	100	.100	.030	.100	.090	**-.010**
1	10	**2**	100	.100	.020	.100	.040	**-.060**
1	10	**1**	100	.100	.010	.100	.010	**-.090**
1	10	**0**	100	.100	.000	.100	.000	**-.100**

そうすると，y が 4 と 3 の間で差がなくなり，その後は x の比例得点のほうが大きくなっていることがわかります。このように，絶対頻度(x <= y)，相対得点(RS(x) >= SF(y))と，比例得点(PS(x) < ~ > PS(y))の大小関係は常に等しいわけではありません。x と y の比例得点が等しくなるときの y の値は，x/a - y/b = 0 から y = x $\sqrt{b/a}$ を導いて数値を代入すると，y = 1 * $\sqrt{100/10}$ = 3.162 になります。

■中世スペイン語の語末母音 e の脱落

スペイン語史の初期に，語末の母音-e は歯・歯茎の単子音 C の後で規則的に脱落しましたが(*ciudad(e), papel(e), mes(e),* etc.)，中世スペイン語のある時期に *present(e), veint(e), adelant(e), part(e), est(e), end(e)*のような 2 子音連続(CC)の後でもしばしば脱落しました。次は，およそ 1500 の公証文書におけるこれら 6 語の語末母音-e の脱落(-CC)と保持(-CCe)の回数を示します。

年代	-CC	-CCe	計	-CC の率	比例得点(PS)
1075	2	2	4	.500	1.0
1100	7	5	12	.583	4.1
1150	15	5	20	**.750**	**11.3**
1175	10	15	25	.400	4.0
1200	25	68	93	.269	6.7
1225	70	173	243	.288	20.2
1250	101	361	462	.219	22.1
1275	228	605	833	.274	62.4
1300	137	418	555	.247	33.8
1325	165	315	480	.344	56.7
1350	102	358	460	.222	22.6
1375	189	312	501	.377	71.3

1400	239	623	862	.277	66.3
1425	52	283	335	.155	8.1
1450	74	535	609	.122	9.0
1475	48	374	422	.114	5.5
1500	45	749	794	.057	2.6
1525	7	386	393	.018	.1
1550		304	304		

この表の比例得点を見ると，-CC の出現がとくに 13 世紀後半から 14 世紀にかけて顕著になっていることがわかります。従来の研究では 13 世紀前半に多かった，と報告されていますが，公証文書ではその時期を後に移動する必要があります。

上の表の 1150 の年代では，-CC の率は確かに.750 と高いのですが，この時期はラテン語の文書が多くスペイン語の資料は限られているので，比例得点はかなり低くなります(11.3)。

■現代スペイン語の文法標識付与

スペイン語などの屈折型ヨーロッパ言語の分析のためには，テキスト中に出現形（変化形）から代表形（名詞ならば単数形，形容詞ならば男性単数形，動詞ならば不定詞）を導き，出現形の文法特徴（品詞，名詞の性・数，動詞の法・時制・人称）を分析しなければなりません。短文ならば人手でも分析できますが，たとえばセルバンテス『ドン・キホーテ』の全文を分析することは無理です。そこで，統計学を利用した「文法標識付与プログラム」(grammatical tagger)を作成します。

変化形が常に一意的な対応があるならば，プログラムは対応表を参照しながら出現形に文法標識を与えればよいので簡単です。たとえば

Alicia estaba durmiendo. (アリシアは眠っていた)

出現形	代表形	文法特徴
Alicia	Alicia	固有名詞
estaba	estar	直説法・線過去・3 人称
durmiendo	dormir	現在分詞

ところが，たとえば pienso という語形には，動詞 pensar「考える」の直説法現在形 1 人称「私が考える」と，男性名詞「肥料」という 2 つの分析が可能です。ここでは el pienso que compró ayer「彼が昨日買った飼料」を分析例とします。はじめにプログラムはすべての品詞の可能性と，前文脈・中文脈・後文脈における品詞連続の頻度(*)を列挙します。

(1) 前文脈: *el pienso {que}*: (58)

«定冠詞 - 名詞 -{関係代名詞}» (54)
«定冠詞 - 名詞 -{接続詞}» (4)
«定冠詞 - 動詞 -{関係代名詞}» (0)
«定冠詞 - 動詞 -{接続詞}» (0)

(2) 中文脈: *pienso {que} compró*: (105)
«名詞 -{関係代名詞}- 動詞» (82)
«名詞 -{接続詞}- 動詞» (23)

(3) 後文脈: *{que} compró ayer*: (16)
«{関係代名詞}- 動詞 - 副詞» (2)
«{接続詞}- 動詞 - 副詞» (14)

そして，(1)前文脈で 4 つの可能性の中から括弧内の頻度が一番高い«定冠詞 - 名詞 -{関係代名詞}» (54)の確率を計算します。54 / (54 + 4 + 0 + 0) = .931。 同様にして，(2)中文脈と(3)後文脈でも最大頻度の確率を求めます。«名詞 -{関係代名詞}- 動詞» (82): 82 / (82 + 23) = .780; «{接続詞}- 動詞 - 副詞»: 14 / (2 + 14) = .875。3 つの文脈の中で最大の確率を示すのは(1)前文脈(.931)になるので，プログラムはこの分析を提示します。

この分析は正しいのですが，ここで仮に(3)の«{関係代名詞}- 動詞 - 副詞»の頻度が 2 ではなくて 1 であった，としましょう。そうすると，«{接続詞}- 動詞 - 副詞»の確率は 14 / (14 + 1) = .933 になりますから，3 つの文脈の中で最大になり，プログラムはこの誤った分析を提示するはずです。

しかし，私たちの直感では，54 / 58 のほうが 14 / 15 よりも重要度が高い，と思われます。先ほどの打席と安打数の関係（打率）を思い出してください。そこで，確率などの率（割合）の軽重を比較するとき比例得点を使えば，$54^2 / 58 = 50.3$ のほうが $14^2 / 15 = 13.1$ よりも大きくなります。次のプログラムは品詞連続の決定のために比例得点を計算しています。

LEXIS-web: Program for lexical analysis of Spanish

Input: Textbox-2; Execution time: 0.179 s. /
Output lines: 5 /

Op	Palabra	C.S.(palabra)	Lema	C.S.(lema)	N.P.	Máx.secuencia	Frec.	Prob.	Ip
1	el	L:ms	el	L	1:L	{L}-Sus-Rel.pro	54	0.93	1
2	pienso	Sus:ms	pienso	Sus	2:Sus;V	L-{Sus}-Rel.pro	54	0.93	1
3	que	Rel.pro	que	Conj#Rel.pro	2:Conj;Rel.pro	Sus-{Rel.pro}-V	82	0.78	1
4	compró	V:IndPas3	comprar	Inf	1:V	Sus-Rel.pro-{V}	82	1.00	1
5	ayer	Adv	ayer	Adv#Sus	2:Adv;Sus	Rel.pro-V-{Adv}	2	0.50	1

http://lecture.ecc.u-tokyo.ac.jp/~cueda/lexis/

5.6. 限定得点

実測値の最小値を 0 とし，最大値を 1 として，範囲を[0.0 ~ 1.0]に限定して計算した値を「限定得点」(Limited Score: LS)と呼びます。次のような行，列，全体の最小値と最大値を使います。

```
Ab2(D,min,max,0,0,0) #データ＋最大値＋最小値
    c1 c2 c3 c4 c5 min max
r1  10 19 14  7 12   7  19
r2  11  7 10  0  1   0  11
r3   0  0  1 12  1   0  12
r4   0  1  2  3  3   0   3
min  0  0  1  0  1   0   -
max 11 19 14 12 12   -  19
```

たとえば r1:c1 (=10)は，{10, 19, 14, 7, 12}という行データの範囲 19-7 = 12 の中で，最小値(=7)から 3 進んだ位置にあります。そこで(10-7) / (19-7) = 3 / 12 = .250 という計算をすると，10 が全体{10, 19, 14, 7, 12}の中で，25% の位置にあることがわかります。

LS = (D-Min) / (Max-Min)
LS: 0.0 (X=Min) ≦ 0.5 (X=(Max-Min) / 2) ≦ 1.0 (X=Max)

ここで Min がデータの最小値，Max がその最大値を示します。X = Min のとき，LS は最小値 0.0 になり，X = Max のとき，L.S.は最大値 1.0 になります。中間点(0.5)は X が Max と Min の中間にあるときです。

(1) 行限定得点と列限定得点

行限定得点(LSr)と列限定得点(LSc)の式は次のようになります。

LSr = (D - MinH) / (MaxH - MinH)
LSc = (D - MinV) / (MaxV - MinV)

ここで MinH は行最小値の縦ベクトル，MinV.は列最小値の横ベクトル，MaxH は行最大値の縦ベクトル，MaxV.は列最大値の横ベクトルを示します。

```
X=Ar2(D,max,min,0,0,0); Y=R(LS(D,1),3); Bind(X,Y) #データ：限定得点(行)
   c1 c2 c3 c4 c5 max min :    c1    c2    c3    c4    c5
r1 10 19 14  7 12  19   7 :  .250 1.000  .583  .000  .417
r2 11  7 10  0  1  11   0 : 1.000  .636  .909  .000  .091
r3  0  0  1 12  1  12   0 :  .000  .000  .083 1.000  .083
r4  0  1  2  3  3   3   0 :  .000  .333  .667 1.000 1.000
```

```
X=Ac2(D,max,min,0,0,0); Y=R(LS(D,2),3); Bind(X,Y) #データ：限定得点: 列
    c1 c2 c3 c4 c5 :    c1    c2    c3    c4    c5
r1  10 19 14  7 12 :  .909 1.000 1.000  .583 1.000
r2  11  7 10  0  1 : 1.000  .368  .692  .000  .000
r3   0  0  1 12  1 :  .000  .000  .000 1.000  .000
r4   0  1  2  3  3 :  .000  .053  .077  .250  .182
max 11 19 14 12 12 :
min  0  0  1  0  1 :
```

(2) 全限定得点

「全限定得点」(Limited Score in all: LSa)の式では，行列全体の最小値 MinA と最大値 MaxA を使います。

LSa = (X - MinA) / (MaxA - MinA)

```
X=Aa2(D,max,min,0,0,0); Y=R(LS(D,3),3); Bind(X,Y) #データ：限定得点: 全
    c1 c2 c3 c4 c5 max min :    c1    c2    c3    c4    c5
r1  10 19 14  7 12   -   - :  .526 1.000  .737  .368  .632
r2  11  7 10  0  1   -   - :  .579  .368  .526  .000  .053
r3   0  0  1 12  1   -   - :  .000  .000  .053  .632  .053
r4   0  1  2  3  3   -   - :  .000  .053  .105  .158  .158
max  -  -  -  -  -  19   - :
min  -  -  -  -  -   -   0 :
```

限定得点は，個々の数値がデータ全体の中で，どのように位置づけられるかを知るために有用です。

ユーザー関数:

```
LS=function(D,s){
  Mx=Ap(D,s,max); Mn=Ap(D,s,min); MV(MV(D,Mn,'s'),Mx-Mn)
} #限定得点(s=1:行, 2:列, 3:全)
```

■語彙の文法カテゴリーと出現頻度

次はセルバンテス『ドン・キホーテ』(第一部: 1605, 第二部: 1615)の全出現語彙を文法カテゴリーと頻度のランクによって分類したものです。頻度のランクは出現度数を対数に変換し，さらに限定得点を使って，それを 1（最小頻度）から 10（最大頻度）に分類しました。それぞれのセルには該当する異なり語数を示しています。

Grammatical category (Members) and Rank (1 - 10)

Category / Rank	1	2	3	4	5	6	7	8	9	10	Total
Noun	1656	973	579	349	171	70	10	2			3.810
Verb	631	399	271	183	93	41	16	9	2	1	1.646
Adjective	562	279	191	122	39	25	5	2			1.225
Adverb	55	36	20	17	18	11	8	4			169
Interjection	10	7	3	1		1					22
Numeral	7	8	8	8	1	3	1				36
Demonstrative pronoun	1	2			1	1	1				6
Indefinite pronoun	2	2	1		8	3					16
Interrogative			2	1	2	2	1				8
Personal pronoun tonic		1	1	1	3	2	2	2			12
Preposition		3		1	4	4	1	3	2	3	21
Determinant				4	11	10	5	4	3	2	39
Conjunction		1		1	1	1	4	3		2	13
Unstressed personal pronoun						3	7		3		13
Relative						1	3			1	5

語彙は冠詞や前置詞・接続詞などの「機能語」(Function Word: 一般に高頻度で小数メンバー)と，名詞，形容詞，動詞などの「内容語」(content: word: 一般に低頻度で多数メンバー)に分類されます。しかし，上の表を見ると，機能語であっても比較的低頻度の語があり，また，内容語であっても比較的頻度が高い語があります。そこで，二分される文法カテゴリーと段階的な頻度について，次のように 4 分割をしました。

Lexicon type / Frequency	High Frequency	Low Frequency
Function Words	Grammatical Words	Instrumental Words
Content Words	Common Words	Specific Words

一般に高頻度語は短縮しやすく，また高頻度の不規則変化形が保持されやすい，と言われます。しかし，同じ高頻度語について述べられていることが，一方で語形が短縮し他方で語形の保持というのでは，一見，矛盾しているかのように思われます。

そこで，それぞれのメンバーを調べると，傾向として，語彙の短縮化はむしろ高頻度の機能語(Grammatical Words：強勢アクセントがないため弱化する)で起こりやすく，一方，不規則変化の保持は高頻度の内容語(Common Words：強勢があるので弱化しない)の特徴だということがわかりました。そこで，言語変化の直接的な要因として頻度を考えるのではなく，むしろ語の機能の違いが語彙の頻度や語形の(不)変化を引き起こしている，と考えたほうがよいと思います。

5.7. 比較得点

個々のセルの値（実測値）を平均値，中央値，中間値，最小値，最大値などのデータの「代表値」と比較したものを**比較得点**(Comparative Score: CS)と呼びます。

(1) 平均値比較得点

```
Ar(D,mean,0,1) #データ＋平均値
   c1 c2 c3 c4 c5 mean
r1 10 19 14  7 12 12.4
r2 11  7 10  0  1  5.8
r3  0  0  1 12  1  2.8
r4  0  1  2  3  3  1.8
```

(a) 平均値比較得点 (差)

「平均値差比較得点」(Comparative Score. Mean Difference: CS.MeD)は，それぞれのセルの値(X)の，平均値からの差を示します[2]。

行平均値差比較得点(Comparative Score. Mean Difference in Row: CS.MeD.R)と，列平均値差比較得点(Comparative Score. Mean Difference in Column: CS.MeD.C)は次のようにして求めます。MeH は横平均列，MeV は縦平均行です。

CS.MeD.R = D - MeH
CS.MeD.C = D - MeV

たとえば，d1:v1=10 の行平均値差比較得点は，d1 の行平均が (10+19+14+7+12) / 5 = 12.4 なので，10 - 12.4 = -2.4 になります。

```
X=Ar(D,mean,0,1); Y=R(CS(D,mean,'d',1),1); Bind(X,Y)
#比較得点: 平均値, 差, 行
   c1 c2 c3 c4 c5 mean :   c1   c2   c3   c4   c5
r1 10 19 14  7 12 12.4 : -2.4  6.6  1.6 -5.4 -0.4
r2 11  7 10  0  1  5.8 :  5.2  1.2  4.2 -5.8 -4.8
r3  0  0  1 12  1  2.8 : -2.8 -2.8 -1.8  9.2 -1.8
r4  0  1  2  3  3  1.8 : -1.8 -0.8  0.2  1.2  1.2
```

[2] 「平均値差」は「偏差」(deviation)とよばれています。

```
X=Ac(D,mean,0,1); Y=R(CS(D,mean,'d',2),1); Bind(X,Y)
#比較得点: 平均値, 差, 列
      c1  c2  c3  c4  c5 :   c1   c2   c3   c4   c5
r1    10  19  14   7  12 :  4.8 12.2  7.2  1.5  7.8
r2    11   7  10   0   1 :  5.8  0.2  3.2 -5.5 -3.2
r3     0   0   1  12   1 : -5.2 -6.8 -5.8  6.5 -3.2
r4     0   1   2   3   3 : -5.2 -5.8 -4.8 -2.5 -1.2
mean 5.2 6.8 6.8 5.5 4.2 :
```

全平均値差比較得点(Comparative Score, Mean Difference in All: CS.MeD.A)は行列全体の平均(MeA)を使います。

CS.MeD.A = D - MeA.

```
X=Aa(D,mean,0,1); Y=R(CS(D,mean,'d',3),1); Bind(X,Y)
#比較得点: 平均値, 差, 全
     c1 c2 c3 c4 c5 mean :   c1   c2   c3   c4   c5
r1   10 19 14  7 12    - :  4.3 13.3  8.3  1.3  6.3
r2   11  7 10  0  1    - :  5.3  1.3  4.3 -5.7 -4.7
r3    0  0  1 12  1    - : -5.7 -5.7 -4.7  6.3 -4.7
r4    0  1  2  3  3    - : -5.7 -4.7 -3.7 -2.7 -2.7
mean  -  -  -  -  -  5.7 :
```

(a) 平均値比較得点 (比)

平均値比比較得点(Comparative Score, Mean Ratio: CS.MeR)は実測値を平均値で割った値（比）です。それぞれ行(R), 列(C), 両軸(B), 全体(A)の平均値比を見ます。X = 0 のときに最小値 0.0 になり, X = 和 (Sm) のとき, 和(Sm) / 平均(Me) = 個数になります。中点の 1.0 は X = Me のときです。

CS.MeR.R = D / MeH_{n1}

CS.MeR.R: 0.0 (X = 0) ≦ 1.0 (X = MeH_{n1}) ≦ P (X = SumH)

CS.MeR.C. = D / MeV_{1p}

CS.MeR.C.: 0.0 (x = 0) ≦ 1.0 (X = MeV_{1p}) ≦ N (x = SumV)

たとえば, d1:v1=10 の行平均値比比較得点は, d1 の行平均が (10+19+14+7+12) / 5 = 62 / 5 = 12.4 なので, 10 / 12.4 ≒ 0.81 になります。

```
X=Ar(D,mean,0,1); Y=R(CS(D,mean,'r',1),3); Bind(X,Y) #比較得点: 平均値,
比, 行
   c1 c2 c3 c4 c5 mean :    c1    c2    c3    c4    c5
r1 10 19 14  7 12 12.4 :  .806 1.532 1.129  .565  .968
r2 11  7 10  0  1  5.8 : 1.897 1.207 1.724  .000  .172
r3  0  0  1 12  1  2.8 :  .000  .000  .357 4.286  .357
r4  0  1  2  3  3  1.8 :  .000  .556 1.111 1.667 1.667
```

```
Bind(RC(D,mean,1),RC(CS(D,mean,'r',2,3),mean,3))
#比較得点: 平均値, 比, 列
      c1  c2  c3  c4  c5 mean :    c1    c2    c3    c4    c5  mean
r1    10  19  14   7  12 12.4 : 1.905 2.815 2.074 1.273 2.824 2.178
r2    11   7  10   0   1  5.8 : 2.095 1.037 1.481     0 0.235  .970
r3     0   0   1  12   1  2.8 :     0     0 0.148 2.182 0.235  .513
r4     0   1   2   3   3  1.8 :     0 0.148 0.296 0.545 0.706  .339
mean 5.2 6.8 6.8 5.5 4.2  5.7 : 1.000 1.000 1.000 1.000 1.000 1.000
```

全平均値比得点(Comparative Score, Mean Ratio in All: CS.MeR.A)は全体の平均値(MeA)を使います。

CS.MeR.B = 2 D / (MeH + MeV)
CS.MeR.A. = D / MeA

```
X=Aa(D,mean,0,1); Y=R(CS(D,mean,'r',3),3); Bind(X,Y)
#比較得点: 平均値, 比, 全
     c1 c2 c3 c4 c5 mean :    c1    c2    c3    c4    c5
r1   10 19 14  7 12    - : 1.754 3.333 2.456 1.228 2.105
r2   11  7 10  0  1    - : 1.930 1.228 1.754  .000  .175
r3    0  0  1 12  1    - :  .000  .000  .175 2.105  .175
r4    0  1  2  3  3    - :  .000  .175  .351  .526  .526
mean  -  -  -  -  -  5.7 :
```

平均値差得点はデータのスケールによって左右されるで，平均差得点を平均値で割ってデータのスケールに合わせます。これを「平均値差比得点」(Comparative Score. Mean Difference ratio: CS.MeDr.)と名づけます[3]。0.0 は参照値(x = Me)です。

CS.MeDr.R = (D - MeH) / MeH1
CS.MeDr.R: -1 (x=0) ≦ 0.0 (x = MeH1) ≦ SumH-MeH1) / MeH1 (x=SumH)

CS.MeDr.C = (D - MeV) / MeV
CS.MeDr.C: -1 (x=0) ≦ 0.0 (x = MeV_{1p}) ≦ SumV - MeV_{1p}) / Me (x=SumV)

全平均値差比得点(Comparative Score. Mean Difference Ratio in All: CS.MeDr.A)は，行列全体の平均 MeA を使います。

CS.MeDr.A = (D - MeA) / MeA

[3] 東京大学教養学部統計学教室(1991:247)は「差比」を「相対誤差」と呼んでいます。

```
X=Ar(D,mean,0,1); Y=R(CS(D,mean,'dr',1),3); Bind(X,Y)
#比較得点: 平均値, 差比, 行
   c1 c2 c3 c4 c5 mean :     c1     c2     c3     c4     c5
r1 10 19 14  7 12 12.4 : -0.194  0.532  0.129 -0.435 -0.032
r2 11  7 10  0  1  5.8 :  0.897  0.207  0.724 -1.000 -0.828
r3  0  0  1 12  1  2.8 : -1.000 -1.000 -0.643  3.286 -0.643
r4  0  1  2  3  3  1.8 : -1.000 -0.444  0.111  0.667  0.667

X=Ac(D,mean,0,1); Y=R(CS(D,mean,'dr',2),3); Bind(X,Y)
#比較得点: 平均値, 差比, 列
      c1  c2  c3  c4  c5 :     c1     c2     c3     c4     c5
r1    10  19  14   7  12 :  0.905  1.815  1.074  0.273  1.824
r2    11   7  10   0   1 :  1.095  0.037  0.481 -1.000 -0.765
r3     0   0   1  12   1 : -1.000 -1.000 -0.852  1.182 -0.765
r4     0   1   2   3   3 : -1.000 -0.852 -0.704 -0.455 -0.294
mean 5.2 6.8 6.8 5.5 4.2 :

X=Aa(D,mean,0,1); Y=R(CS(D,mean,'dr',3),3); Bind(X,Y)
#比較得点: 平均値, 差比, 全
     c1 c2 c3 c4 c5 mean :     c1     c2     c3     c4     c5
r1   10 19 14  7 12    - :  0.754  2.333  1.456  0.228  1.105
r2   11  7 10  0  1    - :  0.930  0.228  0.754 -1.000 -0.825
r3    0  0  1 12  1    - : -1.000 -1.000 -0.825  1.105 -0.825
r4    0  1  2  3  3    - : -1.000 -0.825 -0.649 -0.474 -0.474
mean  -  -  -  -  -  5.7 :
```

(2) 中央値比較得点

比較基準値を平均値ではなく，中央値，中間値，最小値，最大値，大数平均値，大数最頻値とすることも可能です。以下では，中央値比較得点の結果を載せます。

```
X=Ar(D,median,0,1); Y=R(CS(D,median,'d',1),1); Bind(X,Y)
#比較得点: 中央値, 差, 行
   c1 c2 c3 c4 c5 median :   c1   c2  c3   c4   c5
r1 10 19 14  7 12   12.0 : -2.0  7.0 2.0 -5.0  0.0
r2 11  7 10  0  1    7.0 :  4.0  0.0 3.0 -7.0 -6.0
r3  0  0  1 12  1    1.0 : -1.0 -1.0  .0 11.0  0.0
r4  0  1  2  3  3    2.0 : -2.0 -1.0  .0  1.0  1.0

X=Ac(D,median,0,1); Y=R(CS(D,median,'d',2),1); Bind(X,Y)
#比較得点: 中央値, 差, 列
        c1  c2  c3  c4  c5 :   c1   c2   c3   c4   c5
r1      10  19  14   7  12 :  5.0 15.0  8.0  2.0 10.0
r2      11   7  10   0   1 :  6.0  3.0  4.0 -5.0 -1.0
r3       0   0   1  12   1 : -5.0 -4.0 -5.0  7.0 -1.0
r4       0   1   2   3   3 : -5.0 -3.0 -4.0 -2.0  1.0
median 5.0 4.0 6.0 5.0 2.0 :
```

```
X=Aa(D,median,0,1); Y=R(CS(D,median,'d',3),1); Bind(X,Y)
#比較得点: 中央値, 差, 全
       c1 c2 c3 c4 c5 median :   c1   c2   c3   c4   c5
r1     10 19 14  7 12      - :  7.0 16.0 11.0  4.0  9.0
r2     11  7 10  0  1      - :  8.0  4.0  7.0 -3.0 -2.0
r3      0  0  1 12  1      - : -3.0 -3.0 -2.0  9.0 -2.0
r4      0  1  2  3  3      - : -3.0 -2.0 -1.0  0.0  0.0
median  -  -  -  -  -    3.0 :

X=Ar(D,median,0,1); Y=R(CS(D,median,'r',1),3); Bind(X,Y)
#比較得点: 中央値, 比, 行
   c1 c2 c3 c4 c5 median :    c1    c2    c3     c4    c5
r1 10 19 14  7 12   12.0 :  .833 1.583 1.167  0.583 1.000
r2 11  7 10  0  1    7.0 : 1.571 1.000 1.429  0.000  .143
r3  0  0  1 12  1    1.0 :  .000  .000 1.000 12.000 1.000
r4  0  1  2  3  3    2.0 :  .000  .500 1.000  1.500 1.500
> X=Ac(D,median,0,1); Y=R(CS(D,median,'r',2),3); Bind(X,Y)

#比較得点: 中央値, 比, 列
        c1  c2  c3  c4  c5 :    c1    c2    c3    c4    c5
r1      10  19  14   7  12 : 2.000 4.750 2.333 1.400 6.000
r2      11   7  10   0   1 : 2.200 1.750 1.667  .000  .500
r3       0   0   1  12   1 :  .000  .000  .167 2.400  .500
r4       0   1   2   3   3 :  .000  .250  .333  .600 1.500
median 5.0 4.0 6.0 5.0 2.0 :

X=Aa(D,median,0,1); Y=R(CS(D,median,'r',3),3); Bind(X,Y)
#比較得点: 中央値, 比, 全
       c1 c2 c3 c4 c5 median :    c1    c2    c3    c4    c5
r1     10 19 14  7 12      - : 3.333 6.333 4.667 2.333 4.000
r2     11  7 10  0  1      - : 3.667 2.333 3.333  .000  .333
r3      0  0  1 12  1      - :  .000  .000  .333 4.000  .333
r4      0  1  2  3  3      - :  .000  .333  .667 1.000 1.000
median  -  -  -  -  -    3.0 :

X=Ar(D,median,0,1); Y=R(CS(D,median,'dr',1),3); Bind(X,Y)
#比較得点: 中央値, 差比, 行
   c1 c2 c3 c4 c5 median :     c1     c2    c3     c4     c5
r1 10 19 14  7 12   12.0 : -0.167  0.583  .167 -0.417  0.000
r2 11  7 10  0  1    7.0 :  0.571  0.000  .429 -1.000 -0.857
r3  0  0  1 12  1    1.0 : -1.000 -1.000  .000 11.000  0.000
r4  0  1  2  3  3    2.0 : -1.000 -0.500  .000  0.500  0.500

X=Ac(D,median,0,1); Y=R(CS(D,median,'dr',2),3); Bind(X,Y)
#比較得点: 中央値, 差比, 列
        c1  c2  c3  c4  c5 :     c1     c2     c3     c4     c5
r1      10  19  14   7  12 :  1.000  3.750  1.333  0.400  5.000
r2      11   7  10   0   1 :  1.200  0.750  0.667 -1.000 -0.500
r3       0   0   1  12   1 : -1.000 -1.000 -0.833  1.400 -0.500
r4       0   1   2   3   3 : -1.000 -0.750 -0.667 -0.400  0.500
median 5.0 4.0 6.0 5.0 2.0 :
```

```
> X=Aa(D,median,0,1); Y=R(CS(D,median,'dr',3),3); Bind(X,Y)
#比較得点: 中央値, 差比, 全
       c1 c2 c3 c4 c5 median :     c1     c2     c3     c4     c5
r1     10 19 14  7 12      - :  2.333  5.333  3.667  1.333  3.000
r2     11  7 10  0  1      - :  2.667  1.333  2.333 -1.000 -0.667
r3      0  0  1 12  1      - : -1.000 -1.000 -0.667  3.000 -0.667
r4      0  1  2  3  3      - : -1.000 -0.667 -0.333  0.000  0.000
median  -  -  -  -  -    3.0 :
```

ユーザー関数:

```
CS=function(X,f=mean,m='d',s=1){
  #(f=mean,median,mid; m(ethod)='d'(ifference), 'r'(atio), 'dr(dif.ratio)
  #s=1:行 ;  2:列, 3:全
  C=Ap(X,s,f)
  if(m=='d') MV(X,C,'s') else if(m=='r') MV(X,C,'d') else MV(MV(X,C,'s'),C,'d')
} #比較得点
```

5.8. 標準得点

それぞれの行，列または行列全体を同じスケールとばらつきで評価するには，データの平均が 0 に，標準偏差が 1 になるようにする必要があります。この操作は平均(M)からの差（偏差）を標準偏差(Sd)で割ることで可能になります。この値は「標準得点」(Standard Score: SS)と呼ばれます[4]。

```
Ab2(D,mean,sdp,0,1,1) #データ＋平均値＋標準偏差
       c1   c2   c3   c4   c5 mean sdp
r1     10   19   14    7   12 12.4 4.0
r2     11    7   10    0    1  5.8 4.5
r3      0    0    1   12    1  2.8 4.6
r4      0    1    2    3    3  1.8 1.2
mean 5.2 6.8 6.8 5.5 4.2  5.7
sdp  5.3 7.6 5.4 4.5 4.5       5.7
```

次が**行標準得点**(Standard Score in Row:SSR)と**列標準得点**(Standard Score in Column:SSC)の式です。

SSR = (D - M.R) / Sd.R

SSC = (D - M.C) / Sd.C

[4]「標準得点」は Standarized measure, Z-Score ともよばれます。池田央(1975)『統計的方法 I 基礎』（新曜社）。

```
X=Ar2(D,mean,sdp,0,1,1); Y=R(SS(D,1),3); Bind(X,Y) #標準得点: 行
   c1 c2 c3 c4 c5 mean sdp :     c1     c2     c3     c4     c5
r1 10 19 14  7 12 12.4 4.0 : -0.596  1.638  0.397 -1.340 -0.099
r2 11  7 10  0  1  5.8 4.5 :  1.147  0.265  0.926 -1.279 -1.059
r3  0  0  1 12  1  2.8 4.6 : -0.606 -0.606 -0.389  1.991 -0.389
r4  0  1  2  3  3  1.8 1.2 : -1.543 -0.686  0.171  1.029  1.029

X=Ac2(D,mean,sdp,0,1,1); Y=R(SS(D,2),3); Bind(X,Y) #標準得点: 列
      c1  c2  c3  c4  c5 :     c1     c2     c3     c4     c5
r1    10  19  14   7  12 :  0.903  1.620  1.331  0.333  1.704
r2    11   7  10   0   1 :  1.093  0.033  0.596 -1.222 -0.715
r3     0   0   1  12   1 : -0.998 -0.893 -1.055  1.444 -0.715
r4     0   1   2   3   3 : -0.998 -0.760 -0.872 -0.556 -0.275
mean 5.2 6.8 6.8 5.5 4.2 :
sdp  5.3 7.6 5.4 4.5 4.5 :
```

全標準得点(SSA_{np})は全平均(M.A)と全標準偏差(Sd.A)を使います。

$$SSA = (D - M.A) / Sd.A$$

```
X=Aa2(D,mean,sdp,0,1,1); Y=R(SS(D,3),3); Bind(X,Y) #標準得点: 全
     c1 c2 c3 c4 c5 mean sdp :     c1     c2     c3     c4     c5
r1   10 19 14  7 12    -   - :  0.760  2.351  1.467  0.230  1.114
r2   11  7 10  0  1    -   - :  0.937  0.230  0.760 -1.007 -0.831
r3    0  0  1 12  1    -   - : -1.007 -1.007 -0.831  1.114 -0.831
r4    0  1  2  3  3    -   - : -1.007 -0.831 -0.654 -0.477 -0.477
mean  -  -  -  -  -  5.7   - :
sdp   -  -  -  -  -    - 5.7 :
```

このようにして尺度を，平均が 0，標準偏差が 1 になるように標準化させた値が標準得点です。標準化前の数値をそのまま比較すると絶対的な尺度になり，全データの中での相対的な価値が勘案されていないことになります。一方，標準得点は平均がゼロ，標準偏差が 1 になるように標準化されているので，点数とか温度とか価格とか(キロ)メートルのような単位がなくなります。これにより，異なる概念(単位)の数値の間の関係も標準得点によって数値化できるようになります。

ユーザー関数:

```
SS=function(D,s=1){
  Me=Ap(D,s,mean); Sdp=Ap(D,s,sdp); MV(MV(D,Me,'s'),Sdp)
} #標準得点  (s=1:行 ;  2:列; 3:全)
```

●標準得点の平均と標準偏差

標準得点(*SS*)の平均は 0 になり，標準偏差が 1 になります。はじめに，標準得点(*SS*)の平均 E[*SS*]が 0 になることを確かめます。

$$
\begin{aligned}
E[SS] &= E[(X - m) / Sd] && \leftarrow SS = (X - m) / Sd \\
&= (1 / Sd)\, E(X - m) && \leftarrow E(aX) = a\, E(X) \\
&= (1 / Sd)\, [E(X) - E(m)] && \leftarrow E(X + Y) = E(X) + E(Y) \\
&= (1 / Sd)\, [m - E(m)] && \leftarrow E(X) = m \\
&= (1 / Sd)\, [m - m] && \leftarrow E(m) = m \\
&= 0
\end{aligned}
$$

次に，標準得点(SS)の分散 V[SS]が 1 になることを確かめます。分散が 1 であれば標準偏差（＝分散 $^{1/2}$）も 1 になります。

$$
\begin{aligned}
V[SS] &= V[(X - m) / Sd] && \leftarrow SS = (X - m) / Sd \\
&= (1 / Sd)^2\, V(X - m) && \leftarrow V(a\, X) = a^2\, V(X) \\
&= (1 / Sd)^2\, V(X) && \leftarrow V(X - a) = V(X) \\
&= (1 / Sd)^2\, Sd^2 && \leftarrow V(X) = Sd \\
&= 1
\end{aligned}
$$

標準偏差 SD は分散の根（ルート）ですから，標準得点の標準偏差も 1 となります。このように平均 E(X)と分散 V(X)の基本性質を使うと，数理的証明が完結になります。（→「確率」）

●偏差値

テストでよく使われる**偏差値**(Z)は標準得点(SS)を 10 倍し 50 を足して計算します。

$$Z = 10\, SS + 50$$

そうすると以下のように偏差値(Z)の平均 E(Z)は 50 になり，標準偏差は 10 になります。ここでは分散 V(Z)が 100 になることを確認します。

$$
\begin{aligned}
E(Z) &= E(10\, SS + 50) && \leftarrow Z = 10\, SS + 50 \\
&= E(10\, SS) + E(50) && \leftarrow E(X + Y) = E(X) + E(Y) \\
&= 10\, E(SS) + E(50) && \leftarrow E(a\, X) = a\, E(X) \\
&= 10\, 0 + E(50) && \leftarrow E(SS) = 0 \\
&= 50 && \leftarrow E(50) = 50
\end{aligned}
$$

$$
\begin{aligned}
V(Z) &= E(10\, SS + 50) && \leftarrow Z = 10\, SS + 50 \\
&= V(10\, SS) + V(50) && \leftarrow V(X + Y) = V(X) + V(Y) \\
&= 100\, V(SS) + V(50) && \leftarrow V(a\, X) = a\, V(X) \\
&= 100\, 1 + V(50) && \leftarrow V(SS) = 1 \\
&= 100 && \leftarrow V(50) = 0
\end{aligned}
$$

標準得点によって,せっかく平均 0，標準偏差 1 にして標準化したのに，

偏差値では平均 50，標準偏差 10 にしているのです。これは，私たちが 100 点満点のテストに慣れているためで，そのほうがわかりやすいからでしょう。

5.9. 逸脱得点

行列のそれぞれのセルの値がどれほど特異な逸脱度をもつのかを示す得点を「逸脱得点」(Divergent Score: DS)と呼びます。

```
Ab2(D,mean,sdp,0,1,1) #データ＋平均値＋標準偏差
      c1  c2  c3  c4  c5 mean sdp
r1    10  19  14   7  12 12.4 4.0
r2    11   7  10   0   1  5.8 4.5
r3     0   0   1  12   1  2.8 4.6
r4     0   1   2   3   3  1.8 1.2
mean 5.2 6.8 6.8 5.5 4.2  5.7
sdp  5.3 7.6 5.4 4.5 4.5      5.7
```

逸脱得点(DS)は正規分布累積確率を返す R 関数 NORMDIST(x, av, sd, 1)を使って求めます（→確率）。セルの頻度に対応する正規分布累積確率が非常に低いか，または非常に高いとき，その頻度が「普通ではない」「逸脱度が高い」と考えます。上表の実測値を使って，たとえば(d1:v1)の行逸脱得点 DSr(r1:c1)を求めるには

DSr(r1:c1) = pnorm(x, av, sd) = pnorm(10, 12.4, 4.0) = .276

次は行逸脱得点(DSr)です。

```
X=Ar2(D,mean,sdp,0,1,1); Y=R(DS(D,1),3); Bind(X,Y)
#逸脱得点: 行
   c1 c2 c3 c4 c5 mean sdp :   c1   c2   c3   c4   c5
r1 10 19 14  7 12 12.4 4.0 : .276 .949 .654 .090 .460
r2 11  7 10  0  1  5.8 4.5 : .874 .604 .823 .100 .145
r3  0  0  1 12  1  2.8 4.6 : .272 .272 .348 .977 .348
r4  0  1  2  3  3  1.8 1.2 : .061 .246 .568 .848 .848
```

下の表は列逸脱得点(DSc)です。列逸脱得点の計算では，それぞれの列の平均と標準偏差を使って正規分布累積確率を計算します。

```
X=Ac2(D,mean,sdp,0,1,1); Y=R(DS(D,2),3); Bind(X,Y)
#逸脱得点: 列
      c1  c2  c3  c4  c5 :   c1   c2   c3   c4   c5
r1    10  19  14   7  12 : .817 .947 .908 .631 .956
r2    11   7  10   0   1 : .863 .513 .725 .111 .237
r3     0   0   1  12   1 : .159 .186 .146 .926 .237
r4     0   1   2   3   3 : .159 .224 .192 .289 .392
mean 5.2 6.8 6.8 5.5 4.2 :
sdp  5.3 7.6 5.4 4.5 4.5 :
```

全確率得点(DSa)の確率は行列全体の平均と標準偏差を使って正規分布累積確率を計算します。

```
> X=Aa2(D,mean,sdp,0,1,1); Y=R(DS(D,3),3); Bind(X,Y) #逸脱得点: 全
     c1 c2 c3 c4 c5 mean sdp :   c1   c2   c3   c4   c5
r1   10 19 14  7 12    -   - : .776 .991 .929 .591 .867
r2   11  7 10  0  1    -   - : .826 .591 .776 .157 .203
r3    0  0  1 12  1    -   - : .157 .157 .203 .867 .203
r4    0  1  2  3  3    -   - : .157 .203 .257 .317 .317
mean  -  -  -  -  -  5.7   - :
sdp   -  -  -  -  -    - 5.7 :
```

ユーザー関数:

```
DS=function(X,s=1){ #X:データ行列, s=1:行,2:列,3:全
  Me=Ap(D,s,mean); Sd=Ap(D,s,sdp)
  for(i in 1:nrow(X)){for(j in 1:ncol(X)){
    if(s==1) X[i,j]=pnorm(X[i,j],Me[i,1],Sd[i,1])
    if(s==2) X[i,j]=pnorm(X[i,j],Me[1,j],Sd[1,j])
    if(s==3) X[i,j]=pnorm(X[i,j],Me,Sd)
  }}; X
} #逸脱得点
```

●逸脱度の大きな得点

確率得点のそれぞれのセルを見れば，その正常性・異常性がわかりますが,得点の行数･列数が多くなるとそれを視認することが困難になります。そこで確率が 5% (.05)以下と 95%(.95)以上の確率を「逸脱度が高い」と見なしてアステリスク(*)をつけてマークします。さらに，確率が 1% (.01)以下と 99%(.99)以上の確率を「とくに逸脱度が高い」と見なしてシャープ(#)をつけてマークします。

DSMr	c1	c2	c3	c4	c5
r1	.276	.949	.654	.090	.460
r2	.874	.604	.823	.100	.145
r3	.272	.272	.348	*.977	.348
r4	.061	.246	.568	.848	.848

DSMc	c1	c2	c3	c4	c5
r1	.817	.947	.908	.631	*.956
r2	.863	.513	.725	.111	.237
r3	.159	.186	.146	.926	.237
r4	.159	.224	.192	.289	.392

DSMa	c1	c2	c3	c4	c5
r1	.776	#.991	.929	.591	.867
r2	.826	.591	.776	.157	.203
r3	.157	.157	.203	.867	.203
r4	.157	.203	.257	.317	.317

5.10. 有意得点

行列の成分が比較的少ない頻度を示すとき，その成分の数値が「偶然かもしれない」「あまり意味がない（有意でない）」(accidental, not significant)と思われることがあります。そこで二項分布累積確率を使って行列の成分の有意性をチェックします。次のデータ(S1)とその行和(Sh)を使って説明します。

S1	v1	v2	v3	v4	v5
r1	10	19	14	7	12
r2	11	7	10	0	1
r3	0	0	1	12	1
r4	0	1	2	3	3

横軸	Sh
r1	62
r2	29
r3	14
r4	9

たとえば(r1:v1=10)の有意性(significance)を，二項分布累積確率を返す R 関数 pbinom(x, n, e)を使って計算します。二項分布累積確率は生起回数(x)・全試行回数(n)・期待確率(e)によって決まります（→「確率」）。この累積生起確率を集めた得点を「二項有意得点」(Binomial Significant Score: BSS)と呼びます。

上左表の列数は 5 列なので，それぞれのセルの期待確率(e)は一律に 1/5 (20%)であると考えられます[5]。(r1:v1)と(r1:v2)の累積生起確率(aop)は

BSSr(r1:v1) = pbinom(10,62,1/5) = .280
BSSr(r1:v2) = pbinom(19, 62, 1/5) = .984

上の第 1 式は，期待確率(e)が 1/5 (20%)のとき，(r1:v1)=10 未満の頻度(x)が 62 回(n)の中で生起する確率(op)が.180 (18%)であることを示しています。この確率(aop)は 5%以上かつ 95%以下なので有意性がありません（→後述「二項分布累積確率による有意性判定」）。一方，第 2 式の(r1:v2)の頻度 19 の累積生起確率(aop)は.969 (96.9%)であり，これは 95%を超えるのでかなり有意性があると言えます。

全有意得点(SSa)の計算では，行列全体の総和(=114)を試行回数(n)とし，

[5] たとえば，行和が 15 であれば各セルに期待される値（期待値・平均）は 15 にこの期待確率 1/5 を掛けた値 3 になります。

それぞれのセルの値（頻度）を生起回数(x)とし，1 / (P * N)を期待確率(ep)としています。つまり，セル全体に同じ確率(1 / セルの個数)を想定し，それを期待確率として累積生起確率(aop)を求めます。たとえば SSa(S1:r1:v1)は

BSSa(r1:c1) = pbinom(10, 114, 1 / (5 * 4)) = .972

```
BS=function(D,s=3){
  # Binomial security score. s=1:row/2:col/3:all
  W=D; Rs=RowSums(D); Cs=ColSums(D); sm=Sum(D); nr=NR(D); nc=NC(D)
  for(i in 1:nr){for(j in 1:nc){
    P=c(Rs[i],Cs[j],sm); C=1/c(nr,nc,nr*nc)
    W[i,j]=pbinom(D[i,j],P[s],C[s])
  }}; W
}
```

実行結果：BS.r:row / BS.c:col / BS.a:all

```
Q=c(10,19,14,7,12,11,7,10,0,1,0,0,1,12,1,0,1,2,3,3)
Q=DF(matrix(Q,4,5,T)); rownames(Q)='r'&1:4; colnames(Q)='c'&1:5; Q #data
BD(Q,Rnd(BS(Q,1),3),'Q,BS.r')
  [Q] c1 c2 c3 c4 c5 [BS.r]    c1    c2    c3    c4    c5
1  r1 10 19 14  7 12     r1 0.067 0.878 0.394 0.006 0.191
2  r2 11  7 10  0  1     r2 0.961 0.557 0.914 0.000 0.003
3  r3  0  0  1 12  1     r3 0.018 0.018 0.101 1.000 0.101
4  r4  0  1  2  3  3     r4 0.075 0.300 0.601 0.834 0.834
BD(Rnd(BS(Q,2),3),Rnd(BS(Q,3),3),'BS.c,BS.a')
[BS.c]    c1    c2    c3    c4    c5 [BS.a]    c1    c2    c3    c4    c5
1     r1 0.999 1.000 1.000 0.944 1.000     r1 0.972 1.000 0.999 0.789 0.995
2     r2 1.000 0.844 0.989 0.007 0.118     r2 0.988 0.789 0.972 0.003 0.020
3     r3 0.009 0.002 0.019 1.000 0.118     r3 0.003 0.003 0.020 0.995 0.020
4     r4 0.009 0.019 0.072 0.332 0.549     r4 0.003 0.020 0.072 0.173 0.173
```

●二項分布による比率の検定

統計学で扱われる「二項分布による比率の検定」では，たとえば次のような問題が設定されています（市原 1990: 118）。

> コインを 8 回投げて表がでた回数は 1 回だけであった。このようなことは十分ありうるか。

この問題では二項分布の下側を検定するので，表が 1 回も出ない確率(x=0)と表が 1 回出る確率(x=1)とを足し上げます。

P (x = 0, 1) = pbinom(x, n, p) = pbinom(1, 8, 0.5) = .035

このセクションで扱った有意得点は，このような下側の累積確率を示し

ます。一方，たとえば「コインを 8 回投げて表がでた回数が 7 回でたというようなことは十分ありうるか」という検定をするときは，二項分布の上側を見ます。この確率 P(x = 7, 8)を計算するには，

P (x = 7, 8) = 1 - pbinom(x-1, n, p)
= 1 - pbinom(7-1, 8, 0.5, 1) = .035

上の式の中の関数の第 1 引数が x ではなく，x-1 にする理由は，求める確率が上側なので，直接それを求めることができないため，その下側(0 から x-1 まで)の全確率の和を全体(1.000) から引いて求めるためです。この上側確率は「その回数が有意である」と判断したとき，それが実際には間違えている可能性（危険率 Risk : 偶然に起こりうる確率）を示します。「安全率」(Security)は危険率の補数です。

安全率(Security)＋危険率(Risk) = 1.000

ここで取り上げた頻度表などの有意性を検定するときは，一般にピアソンのカイ二乗検定やフィッシャーの精密検定などが使われますが，それらは数値の分布全体をまとめて，その有意性を問題にしています。私たちの有意得点は，表の分布全体をまとめて見るのではなく，表のそれぞれのセルの数値の有意性を問題にしています。たとえば次の D1 と D2 はどちらもカイ二乗検定の結果は同じになります。

D1:	Y (+)	Y (-)	Sh
X (+)	**a: 25**	b: 12	37
X (-)	c: 15	d: 20	35
Sv	40	32	N: 72

D2:	Y (+)	Y (-)	Sh
X (+)	**a: 20**	b: 12	32
X (-)	c: 15	d: 25	40
Sv	35	37	N: 72

χ^2 = 4.448, *p = .035 < .05

しかし，もし X が原因（教育法）であり，Y が結果（成績向上者）を示すようなデータであれば，私たちの関心は«a» (+, +)のほうが«d» (-, -)よりも大きいはずです。

また，安全率と危険率の第 2 引数(n)を頻度分布の横和(Sh)，縦和(Sv)，総和(N)で区別します[6]。

安全率 = pbinom(x-1, n, e)
危険率 = 1 - pbinom(x-1, n, e)

HS(D1)	Y (+)	Y (-)
X (+)	**a: *.976**	b: .010

VS(D1)	Y (+)	Y (-)
X (+)	a: .923	b: .055

TS(D1)	Y (+)	Y (-)
X (+)	**a: *.958**	b: .034

[6] HS[ij] = S(x[ij], Sh[i], 1/2)
VS[ij] = S(x[ij], Sv[j], 1/2)
TS[ij] = S(x[ij], N, 1/4)

X (-)	c: .155 d: .750	X (-)	c: .040 d: .892	X (-)	c: .171 d: .665

横安全率 (HS), 縦安全率 (VS), 全安全率 (TS)

この表を見ると，先の D1 のカイ二乗(χ^2)の有意性は横安全率，または全安全率の«a»の値が強く働いたためであることがわかります。

参考：市原清志 1990『バイオサイエンスの統計学』東京：南江堂

■中世スペイン語の文字<u>, <b>, <v>

下左表は，中世スペイン語の古文書の資料体から得られた頻度分布表です(AF)。下右表はその安全率(S)を示します。

AF	1200	1250	1300	1350	1400
uoz	**3**	**8**	3	11	6
boz	0	3	8	18	35
voz	0	1	1	23	53
Sm	3	12	12	52	94

S	1200	1250	1300	1350	1400
uoz	*.963	*.981	.181	.019	.000
boz	.000	.181	*.981	.526	.758
voz	.000	.008	.008	.934	#1.000

ここでは次のように縦和を使って安全率を計算してあります。

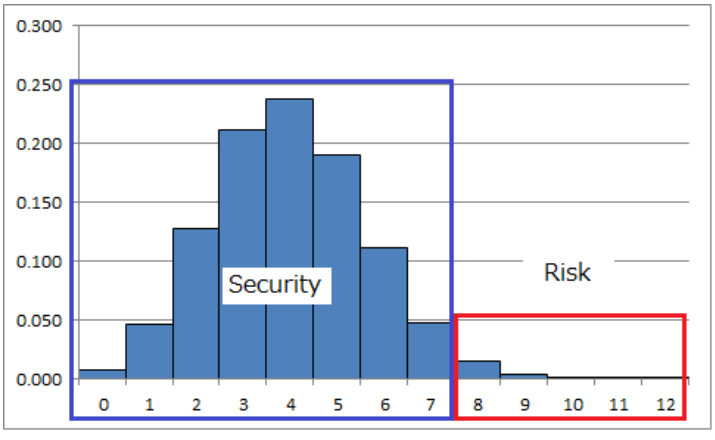
S(3, 3, 1/3) = .963 (96.3%)
S(8, 12, 1/3) = .981 (98.1%)

統計的検定に使われる危険率(p)は，次のように計算されます。

p = 1 - S(3, 3, 1/3) = 1 - .963 = .037 (3.7%)
p = 1 - S(8, 12, 1/3) = 1 - .981 = .019 (1.9%)

次の表と図で安全率(Security)と危険率(Risk)の関係を確認しましょう:

x	Probability	Ac. probability
0	0.008	0.008
1	0.046	0.054
2	0.127	0.181
3	0.212	0.393
4	0.238	0.632
5	0.191	0.822
6	0.111	0.934
7	0.048	0.981
8	**0.015**	**0.996**
9	**0.003**	**0.999**
10	**0.000**	**1.000**



11	**0.000**	**1.000**
12	**0.000**	**1.000**

5.11. 期待確率得点

「３．確率」で扱った「期待確率」を使って，次の期待確率得点行列を出力するプログラムを作成します。

ユーザー関数:

```
EP=function(X,s=1,c=.99){
  #X:データ行列, s=1:行,2:列,3:全, p:母数, c:信頼水準
  W=X; Rs=rowSums(X); Cs=colSums(X); sm=sum(X)
  for(i in 1:nrow(X)){for(j in 1:ncol(X)){
    P=c(Rs[i],Cs[j],sm); W[i,j]=BinE(X[i,j],P[s],c)
  }}; W
} #期待確率得点得点行列
```

実行結果:

```
> X=Ar(D,sum,0,0); Y=Ar(EP(D,1),sum,3,3); Bind(X,Y)
#期待確率得点: 行
   c1 c2 c3 c4 c5 sum :    c1    c2    c3    c4    c5   sum
r1 10 19 14  7 12  62 :  .069  .179  .115  .039  .092  .494
r2 11  7 10  0  1  29 :  .182  .086  .157  .000  .000  .425
r3  0  0  1 12  1  14 :  .000  .000  .001  .522  .001  .523
r4  0  1  2  3  3   9 :  .000  .001  .017  .053  .053  .125

> X=Ac(D,sum,0,0); Y=Ac(EP(D,2),sum,3,3); Bind(X,Y)
#期待確率得点: 列
    c1 c2 c3 c4 c5 :    c1    c2    c3    c4    c5
r1  10 19 14  7 12 :  .226  .463  .289  .116  .397
r2  11  7 10  0  1 :  .264  .093  .169  .000  .001
r3   0  0  1 12  1 :  .000  .000  .000  .288  .001
r4   0  1  2  3  3 :  .000  .000  .006  .021  .027
sum 21 27 27 22 17 :  .490  .556  .464  .425  .426
```

```
X=Aa(D,sum,0,0); Y=Ab(EP(D,3),sum,3,3); Bind(X,Y)
#期待確率得点: 全
     c1 c2 c3 c4 c5 sum :    c1    c2    c3    c4    c5   sum
r1   10 19 14  7 12   - :  .037  .094  .061  .021  .049  .262
r2   11  7 10  0  1   - :  .043  .021  .037  .000  .000  .101
r3    0  0  1 12  1   - :  .000  .000  .000  .049  .000  .049
r4    0  1  2  3  3   - :  .000  .000  .001  .004  .004  .009
sum   -  -  -  -  - 114 :  .080  .115  .100  .073  .053  .421
```

■中世子音字の変異<u>, <b>, <v>の期待確率

先に見た中世子音字の変異 *boz*, *uoz*, *voz* の頻度分布表(下左表: F)から，その期待確率を計算します（下左表: E）。

F	1200	1250	1300	1350	1400
boz	0	3	8	18	35
uoz	3	8	3	11	6
voz	0	1	1	23	53
Sum	3	12	12	52	94

E: *s* = .99	1200	1250	1300	1350	1400
boz	0.000	0.039	0.302	0.200	0.259
uoz	0.215	0.302	0.039	0.097	0.019
voz	0.000	0.001	0.001	0.282	0.439

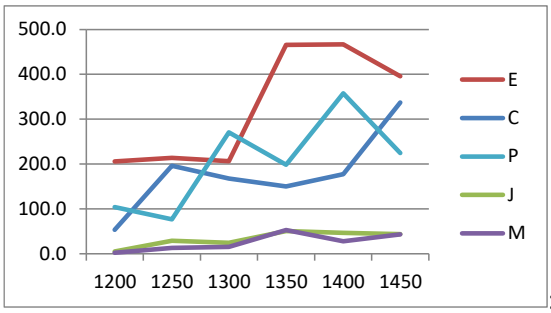
期待確率の表(E)を見ると，*uoz*: 1200 の 3/3 (3 回の試行中 3 回の出現)の期待確率は.215 (21.5%)であり，3/3 = 100%からはかなり遠いことがわかります。uoz:1250 の 8/12 の期待確率(.302)は，uoz:1200 の 3/3 の期待確率(.215)よりも大きくなります。分母が大きい voz:1400 の期待確率.439 は相対頻度 53/94=.564 にかなり近づきます。

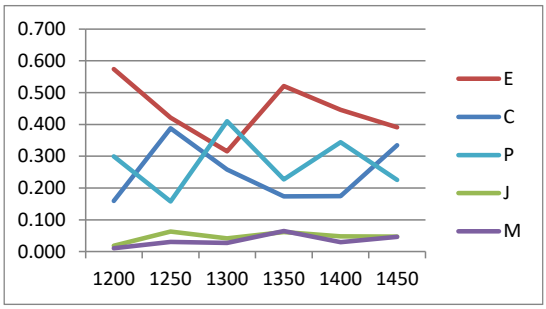
■中世スペイン語古文献の平均語数の期待確率

先に見た中世スペイン語古文献の平均語数(Av)について，その期待確率(E, s=.99)を計算します。このとき，平均語数は小数点を含む連続数なので二項分布確率ではなく，正規分布確率を使用します。

Av.	1200	1250	1300	1350	1400	1450
C	53.8	196.4	168.0	150.3	177.7	337.5
E	205.8	213.8	206.4	465.7	466.7	395.5
J	5.2	29.4	24.9	50.6	46.6	43.9
M	2.7	13.2	15.6	53.3	27.8	43.6
P	104.4	76.8	270.9	198.4	357.8	224.6
Sum	371.9	529.5	685.8	918.3	1076.6	1045.0

E	1200	1250	1300	1350	1400	1450
C	.159	.388	.258	.174	.174	.335
E	.574	.421	.315	.520	.446	.391
J	.018	.063	.042	.061	.048	.047
M	.010	.030	.027	.064	.030	.047
P	.299	.157	.410	.227	.344	.225



:


Av. : (E, s=.99)

平均値(Av)と期待確率(E, s=.99)のグラフを見ると教会文書(E)が平均値では上昇傾向を示し，期待確率では下降傾向を示しています。また平均値で上昇傾向を示している王室文書(C)は，期待確率ではとくに上昇傾向が見えません。平均値はそれぞれの文書の種類の中だけで計算され，一方，期待確率は年代ごとの全体の中での確率を求めているので，このように傾向が一致しないことがあります。

5.12. 確率頻度

たとえば 10 回の試行の中で 3 回出現する事象が m:100 回中平均で何回起こることなのかを 0.99 の有意率(s)で考えます。そこで，先に出現数(x)，試行数(n)，有意率(s)から求めた期待確率(e)を使って，その期待確率のもとで試行数を m にしたときの出現数を求め，これを「確率頻度」(Probabilistic Frequency: PF)と呼びます[7]。

PF = e * m

上の期待確率 e は先述の関数 BinE(x, n, s)で求めます。よって

PF(x, n, s) = e * m = BinE(x, n, s) * m
PF(3, 10, 0.99, 100) = BinE(3, 10, 0.99) * 100 = 0.0475 * 100 = 4.75

この確率頻度を使った行列全体の得点を「確率頻度得点」(Probabilistic Frequency Score: PFS)と呼びます。下左表はそれぞれの年代の文書で見つかった語形の絶対頻度(AF)と列和(Sv)を示し，下右表(PFS:100)はその確率得点(PF)を示します。[uoz:1200]の相対頻度は 3/3 * 100 = 100 となり，最大値になりますが，その確率頻度は 2.2 となって確率的な見地から数値を比較できるようになりました（→後述「各種頻度の問題点」）[8]。

[7] 乗数(m)は自由ですが，データの規模を勘案して，なるべく絶対頻度の桁数と同じにします。
[8] 確率頻度は頻度の一種なので本来ならば整数でよいのですが，正確を期して小数点以下 1 位までを示します。

AF	1200	1250	1300	1350	1400
uoz	3	8	3	11	6
voz	0	1	1	23	53
boz	0	3	8	18	35
Sv	3	12	12	52	94

PFv:10	1200	1250	1300	1350	1400
uoz	2.2	3.0	.4	1.0	.2
voz	.0	.0	.0	2.8	4.4
boz	.0	.4	3.0	2.0	2.6

上右表では試行数(n)として列和(Sv)を使っていますが，下左表は絶対頻度と全語数を示し，下右表はその確率得点(PFS)を示します。

AF	1200	1250	1300	1350	1400
uoz	3	8	3	11	6
voz	0	1	1	23	53
boz	0	3	8	18	35
words	7736	36052	40957	64999	96059

PFv:10^5	1200	1250	1300	1350	1400
uoz	5.6	3.0	.4	1.0	.2
voz	.0	.0	.0	2.8	4.4
boz	.0	.4	3.0	2.0	2.6

最後に相対頻度(RF)・正規化頻度(NF)・確率頻度(PF)の式を比較します。

RF = x / s * m　　s: 和
NF = x / w * m　　w: 語数または文字数
PF = e(x, s, 0.99) * m　　e: 期待確率
(x:出現数, m:乗数)

どれも確率 * 乗数 = 平均値となります。しかし相対頻度(RF)と正規化頻度(NF)では乗数 m を掛ける被乗数が単純に全体の中である事象が起きた割合にしているのに対し，確率頻度(PF)では被乗数が期待確率(e)であることが異なります。

先に示した打者と実験の例についての確率頻度はそれぞれ次のようになります[9]。

事象/有意率	有意率 95%	有意率 99%
a. 10 打席 3 安打の打者の 100 打席の予想安打数	8.7	4.7
b. 100 打席 28 安打の打者の 100 打席の予想安打数	20.7	18.1

このように，予想の有意率を緩めて低くすれば(95%)，それぞれの予想値は楽観的に比較的高くなります。逆に予想の有意率を厳しくして高くすれば(0.99, 99%)，予想値は厳密に低く抑えられます。なお，この予想安打数は一定の有意率で計算された期待確率に厳密に変化がないとしてそれを 100 打席にして計算した平均値です。

NUMEROS.xlsm では次のような行列を入力行列とします。

[9] それぞれ次の関数を使いました。
a: (95%) =PF(3,10,0.95,100), (99%) =PF(3,10,0.99,100)
b = (95%) =PF(28,100,0.95,100), (99%) =PF(28,100,0.99,100)

M	A	B	C	D	E
r1	10	19	14	7	12
r2	11	7	10	0	1
r3	0	0	1	12	1
r4	0	1	2	3	3

この場合「縦軸」を選択します。

実行結果：

PFv:10	A	B	C	D	E
r1	2.3	4.6	2.9	1.2	4.0
r2	2.6	.9	1.7	.0	.0
r3	.0	.0	.0	2.9	.0
r4	.0	.0	.1	.2	.3

●乗数 1 の確率頻度

確率頻度を求める PF(x, n, m, s) '確率頻度(x 出現数,n 試行数,m 乗数,s 有意率)の乗数 m を 1 にすると，期待確率 e そのものが返されます。

PF(x, n, m, s) = BinE(x, n, s) * m

次の実験の結果を見ると，c.10 回の中で 9 回成功した実験(実績成功率: 90%)よりも，d. 100 回の中で 80 回成功した実験(実績成功率: 80%)のほうが，1 回の期待確率（予想成功率）が高いことがわかります。

事象/有意率	有意率. 95	有意率. 99
c. 10 回の中で 9 回成功した実験の 1 回の予想成功率	.606	.496
d. 100 回の中で 80 回成功した実験の 1 回の予想成功率	.723	.691

このように試行回数 n が重要な変数になるので，たとえばある薬草の効果を調べるために 10 回から 100 回まで試験的に使用してそれぞれが 90% の割合で効果があった，と仮定して，それぞれの試験回数における期待確率を見ることにします。

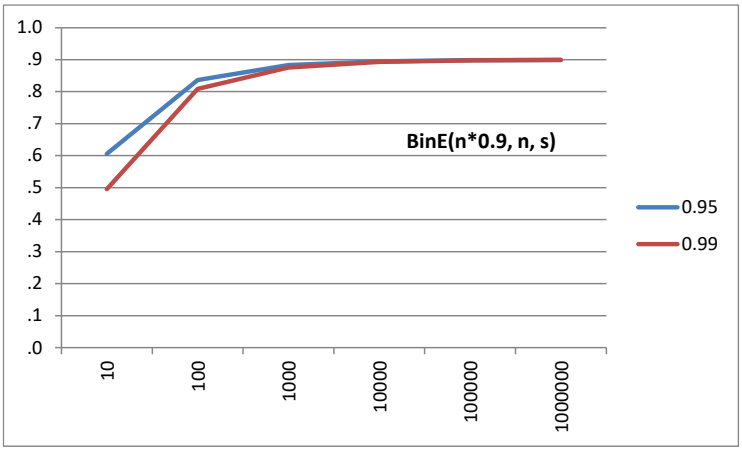
BinE(n*0.9, n, s)	.95	.99
10	0.606	0.496
20	0.717	0.642
30	0.761	0.702
40	0.786	0.736
50	0.801	0.758
60	0.812	0.774
70	0.820	0.786
80	0.827	0.795
90	0.832	0.803
100	0.836	0.809

PF(10*0.9, 10, 0.95) = .606

上の表を見てわかることは，ある薬草の効用の期待確率が 90%である，と言われても，その薬草の試験回数がわずかに 10 回であれば，その中で 9 回で効用があったとしても 0.95 の有意率で言えば，次の 1 回の薬草の効用において成功するのは 60.6%の期待確率でしかないのです。さらに 99%の有意率まで求めれば 49.6%になって，半分以下の予想成功率になってしまいます。期待確率は薬草の試験回数を増やすほど上昇しますが，100 回にしても 90%に達しません。

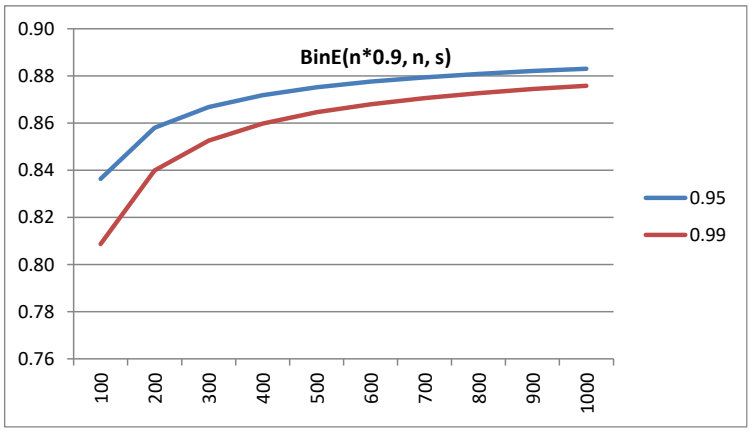
期待確率を上げるには，さらに薬草の試験回数(n)を上げることと，薬草の効用の実績(x)を上げることが考えられます。はじめに薬草の試験回数を 10, 100, 1000 のように増やしていくと，次の表とグラフが示すように期待確率が 90%に近くなるのは少なくとも 1000 回の薬草の効用実績が必要になります。

BinE(n*0.9, n, s)	95%	99%
10	0.6058	0.4957
100	0.8363	0.8087
1000	0.8830	0.8758
10000	0.8949	0.8928
100000	0.8984	0.8978
1000000	0.8995	0.8993



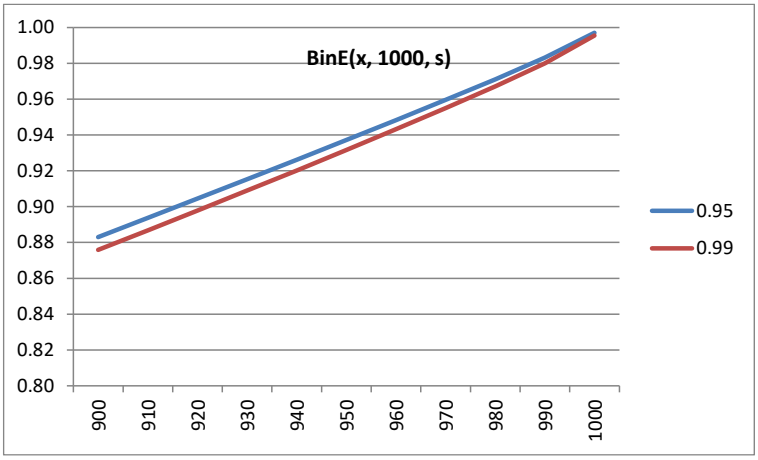
ここで n=100, 200, …, 1000 について期待確率の動きを見ると，はじめに急上昇しますが，次第に上昇が緩慢になることがわかります。

BinE(n*0.9, n, s)	95%	99%
100	0.8363	0.8087
200	0.8580	0.8399
300	0.8668	0.8526
400	0.8718	0.8598
500	0.8751	0.8645
600	0.8775	0.8680
700	0.8794	0.8706
800	0.8808	0.8727
900	0.8820	0.8744
1000	0.8830	0.8758



次に薬草の効用回数を 1000 回として，その成功数を 900, 910, …, 1000 回とすると，次の 1 回の薬草の効用の期待確率はそれぞれ次のようになります。

BinE(x, 1000, s)	95%	99%
900	0.8830	0.8758
910	0.8937	0.8868
920	0.9044	0.8978
930	0.9153	0.9090
940	0.9262	0.9202
950	0.9371	0.9316
960	0.9482	0.9432
970	0.9595	0.9549
980	0.9711	0.9671
990	0.9831	0.9800
1000	0.9970	0.9954



上の表を見ると，1000 回の薬草の効用で 90%成功していれば，それ以上の成功数は期待確率と直線的に相関することがわかります。

以上のことは薬草の各試験で効果あり・効果なしの出現がそれぞれ独立していることを仮定しています。この仮定のもとに効果が未知の薬草を 10 回試用して 9 回に効果があったとしても，その期待確率が 90%であることにはならず，期待確率は 49.5%に留まることになります（99%の有意率）。一方，練習を積んだ運動選手が 10 回のテストで 9 回合格しても期待確率は 49.6%である，次のテストに合格する期待確率が 50%以下である，とは考えられません。練習を積んだ運動選手は，効果が未知の薬草とは異なり，それぞれのテストの合格率はほぼ一定しているはずだからです。

●期待確率・確率頻度の使用についての注意

言語調査において，ある地点で 5 人に質問し，その中の 4 人がある語形を使用したとします。その地点の話者の総人口が 1000 人だとして，99%の有意率でその語形を使用する人の数を推定すると

PF(4, 5, 0.99, 1000) = 222.1

となります。これは総人口(1000 人)のわずか 22.2%に過ぎないので腑に落ちない結果です。さらに別の語形が 5 人中 5 人が使用したとします。同じように確率頻度を計算すると

PF(5, 5, 1000, 0.99) = 398.1

このように全員が使用したとしても 99%の有意率で総人口の 39.8%だけが使用する，ということになります。たとえばキューバで 5 人の話者に聞いて 5 人が全員「バス（交通機関）」を guagua と言う，と回答しても，その言葉 guagua の使用人口が総人口の 39.8%しか想定できないことになるので，やはり変だと思われるでしょう。

共同体の言語使用は言語の共通基盤（コード）に支えられています。よって一人の話者の言語使用は他者の言語使用と強く一致するはずです。ある人が「バス」を guagua と言えば，別の人もやはり「バス」を guagua と言う可能性が非常に高いのです。一方，確率頻度はコインやサイコロを投げることやカードを引くことで観察される二項確率の分布を使って数学的に計算した結果です。たとえば最初に投げたコインが表であっても 2 度目に投げたコインが表になる確率は 1/2 = 0.5 になり，2 つのコインの表と裏の出現は独立して，たがいに関係がありません。

一方，現象のそれぞれの出現が，言語使用のように互いに独立していないと考えられるときには期待確率そのものを予測することには無理があります。野球の打者のそれぞれの打席も完全に独立ではなく，1 回目の安打が 2 回目の安打の確率に影響することはよくあることです。そのようなときには期待確率や確率頻度は「独立した試行であると仮定して」という条件をつけて参考にする程度に留めるべきでしょう。

一方，総数が異なるデータ間で，競合する語形の頻度そのものを評価するときではなく，頻度を比較するときには確率頻度が役立ちます。たとえそれぞれの語形の出現が独立でなくても，完全に同じ条件（出現数・試行数・乗数・有意率）で確率頻度を比較することができるからです。

また，それぞれの試行が独立していなくても，得られた期待確率を検定に使うことは可能です。先述の二項検定では，得られた期待確率が偶然でも起こるとすれば，それはどのような確率になるのかを測り，その有意性を検討するからです。

5.13. 安全得点

言語資料の中に現れる言語現象を絶対頻度・相対頻度・正規化頻度を使って比較するとき，絶対頻度の和や相対頻度・正規化頻度の分母（比較の共通のベース）が小さな値であったり，大きく異なっていたりする場合，

そもそも比較することが困難になります。そこで「３．確率」で，出現数と和・分母から一定の安全率で導き出せる確率を使って求めた値を使う方法を提案しました。この方法は言語資料の分析だけでなく一般の頻度データ分析に応用できます。

「３．確率」で扱った Sf 関数(安全頻度)を使用して次の関数 SF を用意します。

ユーザー関数:

```
SF=function(X, s=1, p=10^3, c=.99){
  #X:データ行列, s=1:行,2:列,3:全, p:母数, c:信頼水準
  W=X; Rs=rowSums(X); Cs=colSums(X); sm=sum(X)
  for(i in 1:nrow(X)){for(j in 1:ncol(X)){
    if(s==1) W[i,j]=Sf(X[i,j],Rs[i],p,c)
    if(s==2) W[i,j]=Sf(X[i,j],Cs[j],p,c)
    if(s==3) W[i,j]=Sf(X[i,j],sm,p,c)
  }}; W
} #安全頻度得点
```

実行結果:

```
X=Ar(D,sum,0,0); Y=Ar(SF(D,1,100),sum,0,0); Bind(X,Y)
#相対得点: 行
   c1 c2 c3 c4 c5 sum : c1 c2 c3 c4 c5 sum
r1 10 19 14  7 12  62 : 14 28 20 10 17  89
r2 11  7 10  0  1  29 : 29 17 26  0  2  74
r3  0  0  1 12  1  14 :  0  0  2 65  2  69
r4  0  1  2  3  3   9 :  0  2  6 12 12  32

> X=Ac(D,sum,0,0); Y=Ac(SF(D,2,100),sum,0,0); Bind(X,Y)
#相対得点: 列
    c1 c2 c3 c4 c5 : c1 c2 c3 c4 c5
r1  10 19 14  7 12 : 34 59 41 21 52
r2  11  7 10  0  1 : 38 18 27  0  2
r3   0  0  1 12  1 :  0  0  2 41  2
r4   0  1  2  3  3 :  0  2  4  7  8
sum 21 27 27 22 17 : 72 79 74 69 64
```

```
> X=Aa(D,sum,0,0); Y=Aa(SF(D,3,100),sum,0,0); Bind(X,Y)
#相対得点: 全
    c1 c2 c3 c4 c5 sum : c1 c2 c3 c4 c5 sum
r1  10 19 14  7 12   - : 10 18 13  7 11   -
r2  11  7 10  0  1   - : 11  7 10  0  1   -
r3   0  0  1 12  1   - :  0  0  1 11  1   -
r4   0  1  2  3  3   - :  0  1  2  3  3   -
sum  -  -  -  -  - 114 :  -  -  -  -  - 110
```

●単純なパーセントの比較

インターネットで商品の好悪の結果を単純にパーセントにして順位を示しているページがあります。そこでは，たとえば 10 人中 8 人が「好き」と答えたアイテム(80%)が，100 人中 55 人が「好き」と答えたアイテム(55%)よりも上位になっていることがあります。この 2 つのケースのそれぞれの期待確率は

e(8, 10, 0.99) = BinE(8, 10, 0.99) = 0.388
e(55, 100, 0.99) = BinE(55, 100, 0.99) = 0.429

よって 100 人中 55 人が「好き」と答えたアイテム(55%)のほうが，10 人中 8 人が「好き」と答えたアイテム(80%)よりも，期待確率(e)が高いのでランクが上になるべきでしょう。

■各種頻度の問題点：スペイン語の文字<b>, <u>, <v>の歴史

(1) 絶対頻度

現代スペイン語の文字 b と v はどちらも同じ音素/b/に対応し発音は同じです。これは歴史的に/v/→/b/という音韻変化があったためです。たとえば voz「声」という語は中世スペイン語(1200-)の公証文書では uoz, voz, boz という 3 つの語形で書かれていました。これらの語形の語頭の u, v, b は voz「声」に限らず，現代スペイン語の語頭に v があるさまざまな語で歴史的に推移しますが，とくに b が摩擦音(v)から閉鎖音(b)に変化した徴候を示しているので注目されます。そこで中世に発行された公証文書中を調べてみると下左表のような絶対頻度(Absolute Frequency: AF)で出現しています。

AF	1200	1250	1300	1350	1400
uoz	3	8	3	11	6
voz	0	1	1	23	53
boz	0	3	8	18	35
Sum	3	12	12	52	94

このような絶対頻度(AF)では，それぞれの年代に発行された文書中の和(Sum)が異なるので年代間の比較ができません。

(2) 相対頻度

絶対頻度(AF)の比較が不可能であることの問題を解決するために下右表(相対頻度 Relative Frequency: RF.100)のように, それぞれの年代の絶対頻度をそれぞれの年代の全数で割り 100 を掛けて列パーセントを計算しました。

AF	1200	1250	1300	1350	1400
uoz	3	8	3	11	6
voz	0	1	1	23	53
boz	0	3	8	18	35
Sum	3	12	12	52	94

RF.100	1200	1250	1300	1350	1400
uoz	100.0	66.7	25.0	21.2	6.4
voz	0.0	8.3	8.3	44.2	56.4
boz	0.0	25.0	66.7	34.6	37.2

これで 1200 - 1400 年代を通じて uoz の割合が次第に減少したこと, voz の割合が次第に増加したこと, そして boz の割合が 1300 年代をピークにして増加・減少したことが確認できるでしょう。

ここで uoz:1200 と uoz:1250 の絶対頻度を見るとそれぞれ 3, 8 であるのに対し, 相対頻度は 100 と 66.7 となって大小関係は大きく逆転しています。しかし単純に相対頻度を比較して「uoz は 1200 年代から 1250 年代になって 33.3 ポイント減少した」と判断することはできません。1200 と 1250 の総和が大きく異なるからです。また boz:1300 は 66.7%よりも voz:1400 (56.4%)のほうが数量的に重要であると思われます。なぜならわずか 12 個の中の 8 個よりも大量の 94 個の中の 53 個のほうが重要な値だと考えられるからです。uoz: 1200 が 100%であるとしても, 全部で 3 個だけのことなのでその頻度をあまり信頼できません。

(3) 正規化頻度

言語データ分析では, それぞれのグループの単語や文字の全数を基盤にした正規化頻度(Normalized Frequency: NF)が使われます。たとえば 1200 年代の文書には全部で 7736 語検出されたので, uoz の 1 万語あたりの正規化頻度は 3 / 7736 * 10000≒3.9 になります。このようにしてすべてのケースの正規化頻度を求めたものが下右表の正規化頻度得点(NF)です。

AF	1200	1250	1300	1350	1400
uoz	3	8	3	11	6
voz	0	1	1	23	53
boz	0	3	8	18	35
words	7736	36052	40957	64999	96059

NF:10^4.	1200	1250	1300	1350	1400
uoz	3.9	2.2	0.7	1.7	0.6
voz	0.0	0.3	0.2	3.5	5.5
boz	0.0	0.8	2.0	2.8	3.6

しかし, この正規化頻度でも先の相対頻度と同様の問題が生じます。たとえば uoz:1200 の 3.9 は boz:1400 の 3.6 よりも大きいことから「boz:1400 は uoz:1200 のレベルに達していない」と判断することはできません。7,736 語の中に占める 3 語を, 96,059 語の中に占める 35 語と簡単に比較することは, 比較の基盤である語数が大きく異なるので(7,736, 96,059), 不可能だ

からです。

(4) 確率頻度

相対頻度も正規化頻度も分母が異なるときに比較するための頻度です[10]。しかし分母が大きく異なるとき，そして一方がとくに小さいときに問題があるのでは，そもそもその用途に疑問が生じます。そこで，分母の違いを考慮して比較するための方法（確率頻度 probabilistic frequency: PF）を考えます。確率頻度によれば 7,736 語の中に占める 3 語は，96,059 語の中に占める 35 語よりも小さな値になります。

AF	1200	1250	1300	1350	1400
uoz	3	8	3	11	6
voz	0	1	1	23	53
boz	0	3	8	18	35
Sv	3	12	12	52	94

PFv:10	1200	1250	1300	1350	1400
uoz	2.2	3.0	.4	1.0	.2
voz	.0	.0	.0	2.8	4.4
boz	.0	.4	3.0	2.0	2.6

■スペイン語の縮約形 del と al

ラテン語から派生した各言語では前置詞と定冠詞が多く縮約します。その中でスペイン語のケースは比較的少なく前置詞 de, a + el の縮約によって del, al だけが形成され，ほかの結合では分離形(de la, a la, en el など)が使われます。スペイン語の歴史研究では「de el, a el が高頻度で用いられたために縮約し del, al となった」と説明されています。しかし古文献資料 CODEA を探しても次の表のように当初(1200 年代)から接合形の del(o)ばかりであって分離形の de el はほとんど見られません（少数の de el は de la などの類推による例外的な表記であって，分離して発音されたとは考えられません）。一方，他の性・数では接合形 delos, dela, delas と分離形 de los, de la, de las が多数見つかりました。

語形/年代.	1200	1300	1400	1500	1600	1700
de el	6	2	0	19	51	16
del	1920	1829	2247	2858	1358	426
de la	309	110	145	370	451	171
dela	957	992	1303	1590	427	171

よって歴史的に見れば de el > del, a el > al というような「高頻度使用による縮約」というプロセスはなかったと思います。Menéndez Pidal (1926-1980: 331)が言うように la, los, las という語形の形成条件が前置詞との接合であったならば，そして，とくに de との接合を考えるならば，当然

[10] 分母が等しいときや非常に近いときならば絶対頻度を使えばよいはずです。

la, los, las という語形が生まれた時以前に dela, delos, delas が存在していたはずだからです。当然 del, al の形成も同時期であったと推定してよいでしょう(de+elo > delo > del; elo > el; a+el > alo > al)。

このような一般的な傾向の概略を探るためには上表のような絶対頻度を比較できません。各年代の歴史資料の全語数が次のように異なるからです。

年代	1200	1300	1400	1500	1600	1700
語数	224,708	230,383	261,564	287,380	125,366	52,938

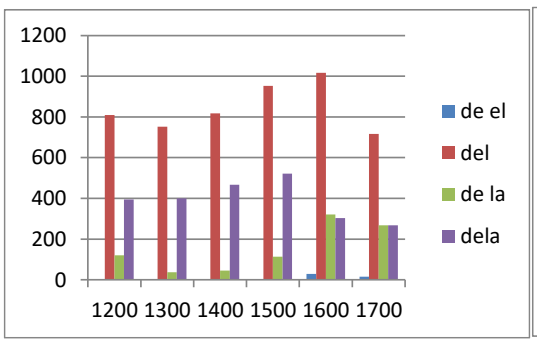
そこで先の絶対頻度表と上の語数表を参照して確率得点を計算すると次のようになります(乗数: 100,000)。

PF: 10^5	1200	1300	1400	1500	1600	1700
de el	.8	.1	.0	3.6	28.6	15.5
del	809.9	751.5	817.6	951.9	1016.4	717.1
de la	120.0	37.8	45.3	113.7	321.6	268.4
dela	394.6	399.5	466.7	521.6	303.5	268.4

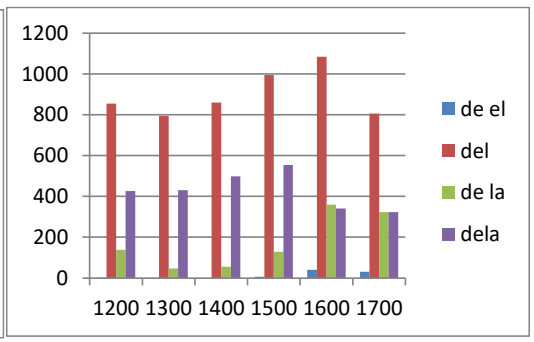
この確率頻度から見ると，さらに de el という分離形がとくに 1500 年代に例外的であったことがわかります(10 万語あたり 3.6)。そして de el という分離形が後年代(1600, 1700)に増えていることは，de el > del という「高頻度使用による縮約」というプロセスの反証になります。「高頻度使用による縮約」があったのならば，分離形の頻度は早期により多くなければならないからです。

さて,この確率頻度によるデータ分析は次の正規化頻度(NF)でも基本的に変わりません。それは 1600 年代，とくに 1700 年代を除けばベースとなる語数に大きな違いがないためです。

NF	1200	1300	1400	1500	1600	1700
de el	2.7	.9	.0	6.6	40.7	30.2
del	854.4	793.9	859.1	994.5	1083.2	804.7
de la	137.5	47.7	55.4	128.7	359.7	323.0
dela	425.9	430.6	498.2	553.3	340.6	323.0



確率頻度 PF



正規化頻度 NF

正規化頻度(NF)と確率頻度(PF)を比べると 1600 年代の de el の頻度では NF が PF よりかなり大きくなり，1700 年では 2 倍近くになっています。しかし先の一般的傾向の観察はほぼ同じです。よって，このように正規化頻度（そして相対頻度）の割り算のベースに大きな違いがなければ，または全体の割り算のベースが十分に大きければ,正規化頻度(そして相対頻度)を使用しても大きな問題は生じません。一方，確率頻度はベースの差を確率で修正しているので，正規化頻度（そして相対頻度）の割り算のベースに対応する試行数(n)に大きな差があっても，そしてもちろん大きな差がなくても，問題は生じません。よって確率頻度は割り算のベースに対応する試行数(n)に大きな差があるときの特殊なケースにも有効な「頑健な」(robust)頻度です。

＊Menéndez Pidal, R. 1926-1980. *Orígenes del español.* Madrid. Espasa-Calpe.

CODEA (Corpus de Documentos Españoles Anteriores a 1800):
http://corpuscodea.es/
確率得点 [2017/4/1]

5.14. 比較期待値得点

ここで提案する 5.14. 比較期待値得点(Comparative Expectation Score: CES)の計算では，次に示す**期待値**(Expected Frequency: Exp)を使います。期待値はそれぞれのセルの値が横の和と縦の和から見て，平均に分布しているとすればどのような値として期待されるか，を示すものです。「期待される」というよりも「予想される」(expected)と考えたほうがわかりやすいかも知れません。

```
X=Ab(D,sum,0,0); Y=Ab(Exp(D),sum,1,1); Bind(X,Y)
#データ行列：期待値行列
    c1 c2 c3 c4 c5 sum :   c1   c2   c3   c4   c5   sum
r1  10 19 14  7 12  62 : 11.4 14.7 14.7 12.0  9.2  62.0
r2  11  7 10  0  1  29 :  5.3  6.9  6.9  5.6  4.3  29.0
r3   0  0  1 12  1  14 :  2.6  3.3  3.3  2.7  2.1  14.0
r4   0  1  2  3  3   9 :  1.7  2.1  2.1  1.7  1.3   9.0
sum 21 27 27 22 17 114 : 21.0 27.0 27.0 22.0 17.0 114.0
```

予想される値「期待値」は横和 Sh と縦和 Sv から計算されます。たとえば，r1 の横和 Sh は 62 です。一方，c1 の縦和は 21 です。総和 S は 114 ですから，r1:c1 は，横和 62 のうち，21 / 114 の割合で出てくると予想されます。つまり，期待値は 62×(21 / 114) ≒ 11.4 となります。期待値行列の行和・列和・総和はデータ行列の行和・列和・総和になります。

$$Exp = (Sr_{n1}\ Sc_{1p}) / St$$

実測値(D)と期待値(Exp)の差(difference: d)，比(ratio: r)，差比(difference ratio: dr)で比較したものを「比較期待得点」(Comparative Expectation Score: CES)と呼びます。それぞれを次の式で導きます。

CESd = D - Exp

CESr = D / Exp

CESdr = (D - Exp) / ExpX=Ab(D,sum,0,0); Y=Ab(CES(D,1),sum,1,1); Bind(X,Y)

```
#比較期待値得点: 差
    c1 c2 c3 c4 c5 sum :   c1   c2   c3   c4   c5 sum
r1  10 19 14  7 12  62 : -1.4  4.3 -0.7 -5.0  2.8  .0
r2  11  7 10  0  1  29 :  5.7  0.1  3.1 -5.6 -3.3  .0
r3   0  0  1 12  1  14 : -2.6 -3.3 -2.3  9.3 -1.1  .0
r4   0  1  2  3  3   9 : -1.7 -1.1 -0.1  1.3  1.7  .0
sum 21 27 27 22 17 114 :   .0   .0   .0   .0   .0  .0

X=Ab(D,sum,0,0); Y=Ab(CES(D,2),sum,1,1); Bind(X,Y)
#比較期待値得点: 比
    c1 c2 c3 c4 c5 sum :  c1  c2  c3  c4  c5  sum
r1  10 19 14  7 12  62 :  .9 1.3 1.0  .6 1.3  5.0
r2  11  7 10  0  1  29 : 2.1 1.0 1.5  .0  .2  4.8
r3   0  0  1 12  1  14 :  .0  .0  .3 4.4  .5  5.2
r4   0  1  2  3  3   9 :  .0  .5  .9 1.7 2.2  5.4
sum 21 27 27 22 17 114 : 2.9 2.8 3.6 6.8 4.2 20.4

X=Ab(D,sum,0,0); Y=Ab(CES(D,3),sum,1,1); Bind(X,Y)
#比較期待値得点: 全
    c1 c2 c3 c4 c5 sum :   c1   c2   c3   c4   c5  sum
r1  10 19 14  7 12  62 : -0.1  0.3  0.0 -0.4  0.3  0.0
r2  11  7 10  0  1  29 :  1.1  0.0  0.5 -1.0 -0.8 -0.2
r3   0  0  1 12  1  14 : -1.0 -1.0 -0.7  3.4 -0.5  0.2
r4   0  1  2  3  3   9 : -1.0 -0.5 -0.1  0.7  1.2  0.4
sum 21 27 27 22 17 114 : -1.1 -1.2 -0.4  2.8  0.2  0.4
```

比較期待得点は全体の期待値と比較するので，軸のオプション（縦軸，横軸，両軸，全体）はありません。

ユーザー関数:

```
Exp=function(X){
  Rs=rowSums(X); Cs=colSums(X); s=sum(X)
  for(i in 1:nrow(X)){for(j in 1:ncol(X)){X[i,j]=Rs[i]*Cs[j]}}; X/s
} #期待値行列 Expectation
```

```
CES=function(D,s=1){ #s=1: 差, 2: 比, 3:差比
  if(s==1) D-Exp(D) else if(s==2) D/Exp(D)else (D-Exp(D))/Exp(D)
} #比較期待値得点 Comparative Expectation Score
```

●特化係数

下のようなクロス集計表のそれぞれのセルの数値を全体の数値と比較する方法として「特化係数」(specialization coefficient)が使われます。次の表(D)は渡辺・神田(2008:172)で想定されたデータ例です。800 人に，現内閣を支持するか，しないかを問い，男性と女性の内訳を示しています。

D	支持	不支持	S
男性	230	250	480
女性	120	200	320
T	350	450	800

下左表(Rr)はそれぞれのセルを横和(S)で割った比率を示します。そして，下右表(Spr)は，それぞれの比率を全体の比率(T)で割った値です。これが特化係数となります。

Spr(男性：支持) = 比率(男性：支持) / 比率(比率) = .479 / .438 = 1.095

Rr	支持	不支持	S
男性	0.479	0.521	1.000
女性	0.375	0.625	1.000
T	0.438	0.563	1.000

Spr	支持	不支持	S
男性	1.095	0.926	1.000
女性	0.857	1.111	1.000
T	1.000	1.000	1.000

このように特化係数はそれぞれの比率を全体の比率と比較します。それぞれの比率が全体の比率より大であれば 1 以上になり，それが小であれば 1 以下になります。上右表を見ると，とくに女性：支持の特化係数が小さいので，この表内では女性固有の特徴であると考えられます。

同様にしてそれぞれの比率と全体の比率を縦和から求めても，次のように同じ結果になります（渡辺・神田 2008:175）。

Rc	支持	不支持	S
男性	0.657	0.556	0.600
女性	0.343	0.444	0.400
T	1.000	1.000	1.000

Spc	支持	不支持	S
男性	1.095	0.926	1.000
女性	0.857	1.111	1.000
T	1.000	1.000	1.000

その理由は次の計算から明らかです。

Spr[ij] = (X[ij] / S[i]) / (T[j] / N) = (X[ij] * N) / (S[i] * T[j])
Spc[ij] = (X[ij] / T[j]) / (S[i] / N) = (X[ij] * N) / (S[i] * T[j])

よって，Spr[ij] = Spc[ij]

ここで X[ij]はデータ，S[i]は横和，T[j]は縦和，N は総和を示します。

上の式を見ると，特化係数は比較期待得点(比: CESr)と同じになることがわかります。

CESr[ij] = X[ij] / E[ij] = X[ij] / (S[i]*T[j]/N) = (X[ij] * N) / (S[i] * T[j])

さて，あらためてデータ行列(D)と特化係数(Spr, Spc)を比べると，データ行列では男性の支持数(230)が男性の不支持数(250)より小さいのに，特化係数では逆にそれが大きくなっています(1.095, 0,026)。また，データ行列の最大値は男性：不支持の 250 ですが，特化係数の最大値は女性：不支持の 1.111 になっています。データ最大値の男性：不支持の特化係数は 1 以下です。このように特化係数はデータ行列の大小関係が反映されていないことがあります。これは特化係数は全体の比率(S, T)を考慮に入れて計算するためです。データ行列の大小関係を反映させるには，次のような「セル比率(C)」（渡辺・神田 2008:178）を使うとよいでしょう。これは「総和による標準化」(→「相対得点」)と同じです。

C[ij] = D[ij] / N

C	支持	不支持	S
男性	0.288	0.313	0.600
女性	0.150	0.250	0.400
T	0.438	0.563	1.000

参考：渡辺美智子・神田智弘 2008『実践ワークショップ Excel 徹底活用統計データ分析』(改訂新版)東京：秀和システム

5.15. 順位得点

数値の多寡を順位で示す「順位得点」(Rank Score)には昇順と降順を区別します。昇順で rank 関数は次のように昇順の順位を返します。昇順では，最小の 7 の順位が 1 となり，次の 10 の順位が 2 となり，最大の 19 の順位は 5 となります

```
rank(c(10,19,14,7,12)) # 2 5 4 1 3
```

降順の順位はデータを負に変えてから(-10,-19,-14,-7,-12)，rank 関数に渡します。その最小値-19 の順位が 1 となります。つまりデータの最大値(19)の順位が 1 となります。

```
rank(-c(10,19,14,7,12)) # 4 1 2 5 3
```

次の出力の左側はデータ行列，中央は昇順・順位得点，右側は昇降順・順位得点を示します。

```
Bind(Bind(D,Rank(D,1,F)),Rank(D,1,T))
#順位得点 tie=min (データ:昇順:降順): 行
   c1 c2 c3 c4 c5 : c1 c2 c3 c4 c5 : c1 c2 c3 c4 c5
r1 10 19 14  7 12 :  2  5  4  1  3 :  4  1  2  5  3
r2 11  7 10  0  1 :  5  3  4  1  2 :  1  3  2  5  4
r3  0  0  1 12  1 :  1  1  3  5  3 :  4  4  2  1  2
r4  0  1  2  3  3 :  1  2  3  4  4 :  5  4  3  1  1

> Bind(Bind(D,Rank(D,2,F)),Rank(D,2,T))
#順位得点 tie=min (データ:昇順:降順): 列
   c1 c2 c3 c4 c5 : c1 c2 c3 c4 c5 : c1 c2 c3 c4 c5
r1 10 19 14  7 12 :  3  4  4  3  4 :  2  1  1  2  1
r2 11  7 10  0  1 :  4  3  3  1  1 :  1  2  2  4  3
r3  0  0  1 12  1 :  1  1  1  4  1 :  3  4  4  1  3
r4  0  1  2  3  3 :  1  2  2  2  3 :  3  3  3  3  2

> Bind(Bind(D,Rank(D,3,F)),Rank(D,3,T))
#順位得点 tie=min (データ:昇順:降順): 全
   c1 c2 c3 c4 c5 : c1 c2 c3 c4 c5 : c1 c2 c3 c4 c5
r1 10 19 14  7 12 : 14 20 19 12 17 :  6  1  2  8  3
r2 11  7 10  0  1 : 16 12 14  1  5 :  5  8  6 17 13
r3  0  0  1 12  1 :  1  1  5 17  5 : 17 17 13  3 13
r4  0  1  2  3  3 :  1  5  9 10 10 : 17 13 12 10 10
```

●平均順位得点

たとえば{1, 4, 4, 4, 10}という頻度の昇順位得点は{1, 2, 2, 2, 5}となります[11]。このように，同順位(2)が続くとき，2, 3, 4 の平均(=3)を求めて，その順位とする方法を「平均順位得点」(Mean Rank Score: MRS)と呼びます。この場合の平均順位得点は{1, 3, 3, 3, 5}となります。

ここで a 番目から b 番目の平均順位(m)は

$$m = (a + b) / 2$$

となります[12]。上の場合は a = 2, b = 4 なので m = (2 + 4) / 2 = 3。この平均順位得点のほうが同順位の数値をより現実的に表現していることは，同順位が多い場合を見るとわかります。たとえば，{1, 4, 4, 4, 4, 4, 10}の順位得点{1, 2, 2, 2, 2, 2, 7}よりも平均順位得点{1, 4, 4, 4, 4, 4, 7}の数値のほうが

---

[11] これは Excel 関数 RANK の方法です。このようなランクは競技や試験などの同点者に対して一律に有利な順位を与えるときに使われます。

[12] 統計学の本では「a 番目から始まる n 個の同順位 r」の式

$$r = (a - 1) + (n + 1) / 2$$

が示されていますが，n = b - a + 1 になるので（b は最後の番），

$$r = (a - 1) + (b - a + 1 + 1) / 2 = (2a - 2 + b - a + 2) / 2 = (a + b) / 2$$

となります。a と b を使う式のほうが簡単で r の数値の意味を直ちに理解できます。

正確です[13]: (2+6)/2 = 4。順位得点を用いた相関係数や検定では，この平均順位得点が使われます。以下のように，平均順位は小数点がつくことがあります。

```
Bind(Bind(D,Rank(D,1,F,'average')),Rank(D,1,T,'average'))
#平均順位得点 (データ:昇順:降順): 行
   c1 c2 c3 c4 c5 :  c1  c2  c3  c4  c5 :  c1  c2  c3  c4  c5
r1 10 19 14  7 12 :   2   5   4   1   3 :   4   1   2   5   3
r2 11  7 10  0  1 :   5   3   4   1   2 :   1   3   2   5   4
r3  0  0  1 12  1 : 1.5 1.5 3.5   5 3.5 : 4.5 4.5 2.5   1 2.5
r4  0  1  2  3  3 :   1   2   3 4.5 4.5 :   5   4   3 1.5 1.5

Bind(Bind(D,Rank(D,2,F,'average')),Rank(D,2,T,'average'))
#平均順位得点 (データ:昇順:降順): 列
   c1 c2 c3 c4 c5 :  c1 c2 c3 c4  c5 :  c1 c2 c3 c4  c5
r1 10 19 14  7 12 :   3  4  4  3   4 :   2  1  1  2   1
r2 11  7 10  0  1 :   4  3  3  1 1.5 :   1  2  2  4 3.5
r3  0  0  1 12  1 : 1.5  1  1  4 1.5 : 3.5  4  4  1 3.5
r4  0  1  2  3  3 : 1.5  2  2  2   3 : 3.5  3  3  3   2

Bind(Bind(D,Rank(D,3,F,'average')),Rank(D,3,T,'average'))
#平均順位得点 (データ:昇順:降順): 全
   c1 c2 c3 c4 c5 :   c1   c2   c3   c4   c5 :   c1   c2   c3   c4   c5
r1 10 19 14  7 12 : 14.5   20   19 12.5 17.5 :  6.5    1    2  8.5  3.5
r2 11  7 10  0  1 :   16 12.5 14.5  2.5  6.5 :    5  8.5  6.5 18.5 14.5
r3  0  0  1 12  1 :  2.5  2.5  6.5 17.5  6.5 : 18.5 18.5 14.5  3.5 14.5
r4  0  1  2  3  3 :  2.5  6.5    9 10.5 10.5 : 18.5 14.5   12 10.5 10.5
```

ユーザー関数:

```
Rank=function(X,s=1,d=T,met='min'){#sel=1:行, 2:列, 3:全; d=T:降順
  if(d) X=-X
  if(s==3){Vc=c(as.matrix(X)); Rk=rank(Vc,ties.method=met)
    W=matrix(Rk, nrow=nrow(X)); rownames(W)=rownames(X); colnames(W)=colnames(X)
  } else W=apply(X,s,function(X){rank(X,ties.method=met)})
  if(s==1) W=t(W); W #横(sel==1)→転置   met:'first','min','average'
} #順位得点 (s=1:行 ;  2:列) d:降順
```

■接触言語の子音文字(2)

先に見たように（→確率得点），ラテンアメリカの先住民言語 Aymara (Bolivia), Gyuarani (Paraguay), Mextec (Mexico)は文字体系の中にスペイン語特有の文字エニェ(ñ)を取り入れています。それぞれの言語の子音文字の中で文字エニェ(ñ)が占める位置を調べるときの1つの方法として以下のよ

[13] このように平均順位得点を使えば，この例の 7 位の者が 2-6 位の者が一律に 2 位として扱われることへの不公平感(自分が不利になるわけではない, にもかかわらず, 他者が有利になることへの不満)がなくなるはずです。

うな降順位得点があります[14]。

C	Aymara	Garifuna	Guarani	Mazateco	Mixteco	Otomí	Quechua	Zapoteco
b	17	6	14	13	17	17	18	4
c	13	16	8	11	10	12	9	14
d	17	5	11	8	8	5	16	5
f	16	14	18	15	21	11	19	20
g	17	3	12	6	16	9	17	11
h	5	8	2	12	12	7	10	15
j	7	18	13	2	6	4	13	16
k	1	15	7	3	2	3	5	2
l	15	2	15	17	13	19	8	9
m	8	7	6	9	11	8	11	8
n	3	1	9	1	1	1	1	1
ñ	12	11	16	16	4	14	15	18
p	6	17	3	19	18	15	4	13
q	10	18	20	19	18	20	2	20
r	9	4	1	14	14	6	12	9
s	4	10	17	7	7	18	3	6
t	2	9	4	4	3	2	7	3
v	17	18	5	18	9	20	21	19
w	11	12	20	19	21	20	14	20
x	17	18	20	5	15	16	22	7
y	14	13	10	10	5	10	6	12
z	17	18	19	19	18	12	19	16

文字エニェ(ñ)は，どの言語でも比較的低い頻度順位を示しますが，メキシコの Mixteco 語では非常に高い順位(n, k, t に続く 4 位)になっています。その語内の出現位置を調べると語頭位置(#ñ-)での頻度が高いことがわかります(184)。文字エニェ(ñ)を使う接触言語の特徴として，語頭位置(#ñ-)に硬口蓋鼻音[ɲ]が現れることがあげられます。例：*ña* (Otomí), *ña-i* (Mixteco), *ñambohasa* (Guarani). これは文字エニェ(ñ)を生成したスペイン語（語中に限

[14] アイマラ語(Aymara, Bolivia)，ガリフナ語(Galifuna, Honduras)，グアラニ語(Guarani, Paraguay)，マサテコ語(Mazateco, Mexico)，ミステコ語(Mixteco, Mexico)，オトミ語(Otomí, Mexico)，ケチュア語(Quechua, Peru)，サポテコ語(Zapoteco, Mexico). 資料は United Nations Human Rights http://www.ohchr.org/EN/Pages/WelcomePage.aspx (2015/11/13)
このサイトでは数種の Quechua 語が記録されていますが，ここではペルーの Cusco のバリアントを使いました。

る）にはなかった特徴です。

Lengua	#ñ&	&ñ&	&ñ#	Total
Aymara	6	160	9	175
Garifuna	14	76		90
Guarani	16	60		76
Mazateco		8		8
Mixteco	184	44	2	230
Otomí	28	38		66
Quechua (Cuzco)	31	19		50
Zapoteco	3			3
Total	282	405	11	698

5.16. 連関得点

後述する各種の**連関係数**(Coefficient of association)を応用して，行と列の連関性を示す得点を**連関得点**(Association Score: AS)と呼びます。連関得点の計算には A_{np}, B_{np}, C_{np}, N_{np} という行列が必要です。A_{np} は行と列がどちらも選択されている個数(+/+)と見なします。これは入力行列 D_{np} と同じです。B_{np} は行が選択され列が選択されていない個数を示し(+/-)，C_{np} は逆に，行が選択されず列が選択されていない個数を示します(-/+)。そして，N_{np} は行も列も選択されていない個数を示します(-/-)。

実測値	c1	c2	c3	c4	c5	**和** Sh
r1	10	19	14	7	12	62
r2	11	7	10	0	1	29
r3	0	0	1	12	1	14
r4	0	1	2	3	3	9
和 Sc	21	27	27	22	17	s: 114

たとえば，r1:c1 の 10 を，r1(+):c1(+)の回数(A:+/+)と見做します。r1(+):c1(-)の回数(B:+/-)は，横和(Sh) - X (10) = 62 - 10 = 52 になります。また，r1(-):c1(+)の回数(C:-/+)は，縦和(Sc) - 10 = 21 - 10 = 11 です。そして r1(-):c1(-)の回数(N:-/-)は，総和(s)から A+B+C を引けば求めることができます(N=114 - (10+52+11) = 41)。

実測値	c1	c2	c3	c4	c5
r1	A:10	B:52			
r2 r3 r4	C:11	N:41			

他の成分についても同様に A, B, C, D の値が求められます。たとえば, r2:c2 については,

実測値	c1	c2	c3	c4	c5
r1	N:10	C:19	N:33		
r2	B:11	A:7	B:11		
r3 r4	N:0	C:1	N:22		

そこで，次の行列を用意します。

$A_{np} = D_{np}$
$B_{np} = Sh_{n1} - D$
$C_{np} = Sv_{p} - D$
$N_{np} = s - A_{np} - B_{np} - C_{np}$

```
A=N ; B=MV(Rsums(N),N,'s')
C=MV(Csums(N),N,'s'); N=MV(MV(MV(sum(N),A,'s'),B,'s'),C,'s')
BIND(A,B,C,N)
   c1 c2 c3 c4 c5 : c1 c2 c3 c4 c5 : c1 c2 c3 c4 c5 : c1 c2 c3 c4 c5
r1 10 19 14  7 12 : 52 43 48 55 50 : 11  8 13 15  5 : 41 44 39 37 47
r2 11  7 10  0  1 : 18 22 19 29 28 : 10 20 17 22 16 : 75 65 68 63 69
r3  0  0  1 12  1 : 14 14 13  2 13 : 21 27 26 10 16 : 79 73 74 90 84
r4  0  1  2  3  3 :  9  8  7  6  6 : 21 26 25 19 14 : 84 79 80 86 91
```

この A, B, C, N という行列を用いて，それぞれのセルに該当する連関係数を求め，これを**連関係数得点**(Association Score: AS)と呼びます。たとえば, 次は単純一致係数(Simple matching coefficient)を使った**単純一致係数得点**(Simple matching score: AS.Sm)を示します。S.m.s.は N 値を重視するため, 全体に数値が高くなる傾向があります。

$AS.Sm = (A_{np} + N_{np}) / (A_{np} + B_{np} + C_{np} + N_{np})$

```
Bind(D,R(AS(D,'s'),3)) #Simple matching
   c1 c2 c3 c4 c5 :   c1   c2   c3   c4   c5
r1 10 19 14  7 12 : .447 .553 .465 .386 .518
r2 11  7 10  0  1 : .754 .632 .684 .553 .614
r3  0  0  1 12  1 : .693 .640 .658 .895 .746
r4  0  1  2  3  3 : .737 .702 .719 .781 .825
```

次は **Jaccarr 係数得点**(AS.J)と **Dice 係数得点**(AS.D)です。

$AS.J = A_{np} / (A_{np} + B_{np} + C_{np})$

$AS.D = A_{np}*2 / (A_{np}*2 + B_{np} + C_{np})$

```
Bind(R(AS(D,'j'),3),R(AS(D,'d'),3)) #Jaccard, Dice
     c1   c2   c3   c4   c5 :   c1   c2   c3   c4   c5
r1 .137 .271 .187 .091 .179 : .241 .427 .315 .167 .304
r2 .282 .143 .217 .000 .022 : .440 .250 .357 .000 .043
r3 .000 .000 .025 .500 .033 : .000 .000 .049 .667 .065
r4 .000 .029 .059 .107 .130 : .000 .056 .111 .194 .231
```

次は **Russel & Rao 係数得点**(RR)と **Russel & Rao-2 係数得点**(RR2)です。

$AS.RR = A_{np} / (A_{np} + B_{np} + C_{np} + N_{np})$

$AS.RR2 = A_{np}*3 / (A_{np}*3 + B_{np} + C_{np} + N_{np})$

```
Bind(R(AS(D,'rr'),3),R(AS(D,'rr2'),3))
#Russel & Rao, Russel & Rao-2
     c1   c2   c3   c4   c5 :   c1   c2   c3   c4   c5
r1 .088 .167 .123 .061 .105 : .224 .375 .296 .164 .261
r2 .096 .061 .088 .000 .009 : .243 .164 .224 .000 .026
r3 .000 .000 .009 .105 .009 : .000 .000 .026 .261 .026
r4 .000 .009 .018 .026 .026 : .000 .026 .051 .075 .075
```

次は **Hamann 係数得点**(AS.H)と **Yule 係数得点**(AS.Y).です。

$AS.H = [(A_{np} + N_{np}) - (B_{np} + C_{np})] / [(A_{np} + N_{np}) + (B_{np} + C_{np})]$

$AS.Y = [(A_{np} * N_{np}) - (B_{np} * C_{np})] / [(A_{np} * N_{np}) + (B_{np} * C_{np})]$

```
 Bind(R(AS(D,'h'),3),R(AS(D,'y'),3)) #Hamann, Yule
       c1   c2     c3     c4   c5 :     c1     c2     c3     c4     c5
r1 -0.105 .105 -0.070 -0.228 .035 : -0.165  0.417 -0.067 -0.522  0.386
r2  0.509 .263  0.368  0.105 .228 :  0.642  0.017  0.356 -1.000 -0.733
r3  0.386 .281  0.316  0.789 .491 : -1.000 -1.000 -0.641  0.964 -0.425
r4  0.474 .404  0.439  0.561 .649 : -1.000 -0.449 -0.045  0.387  0.529
```

次は **Phi 係数得点**(AS.Ph)と **Ochiai 係数得点**(AS.O)です。

$$AS.Ph = [(A_{np} * N_{np}) - (B_{np} * C_{np})] / [(A_{np} + B_{np})*(C_{np}+N_{np})*(A_{np}+C_{np})*(B_{np}+N_{np})]^{1/2}$$

$$AS.O = S_{np} / [(A_{np} + B_{np})*(A_{np}+C_{np})]^{1/2}$$

```
Bind(R(AS(D,'p'),3),R(AS(D,'o'),3)) #Phi, Ochiai
       c1     c2     c3     c4     c5 :   c1   c2   c3   c4   c5
r1 -0.065  0.179 -0.028 -0.222  0.136 : .277 .464 .342 .190 .370
r2  0.294  0.006  0.148 -0.286 -0.188 : .446 .250 .357 .000 .045
r3 -0.178 -0.208 -0.146  0.630 -0.082 : .000 .000 .051 .684 .065
r4 -0.139 -0.087 -0.010  0.104  0.151 : .000 .064 .128 .213 .243
```

それぞれの連関係数の特徴については後述します(→「相関」「連関係数」)。

ユーザー関数:

```
AS=function(X,s='um'){
  nr=NR(X); nc=NC(X); m=nr+nc-2; n=Sum(X)
  A=X ; B=RowSums(X)-X; C=t(ColSums(X)-t(X)); D=n-A-B-C
  dplyr::case_when(
    s=='s' ~ ( A+D)/n, # Simple matching
    s=='d' ~    A*2/(A*2+B+C), # Dice
    s=='dm' ~ A*m/(A*m+B+C), # Dice.m
    s=='j' ~    A/(A+B+C), #Jaccard
    s=='jm' ~ A*m/(A*m+B+C), # Jaccard.m
    s=='r' ~ A/n, #Russel & Rao
    s=='r3' ~ A*3/(A*3+B+C+D), #Russel & Rao-2
    s=='rm' ~ A*m/(A*m+B+C+D), #Russel & Rao-2
    s=='h' ~ ((A+D)-(B+C))/((A+D)+(B+C)), #Hamann
    s=='y' ~ (A*D-B*C)/(A*D+B*C), # Yule
    s=='p' ~ (A*D-B*C)/sqrt((A+B)*(C+D)*(A+C)*(B+D)), # Phi
    s=='o' ~ A/sqrt((A+B)*(A+C)),    # Ochiai
    s=='u2' ~ (A*2-B-C)/(A*2+B+C), # Ueda.2
    s=='um' ~ (A*m-B-C)/(A*m+B+C), # Ueda.m
    s=='mi' ~ log2(A*n/(A+B)/(A+C)), # Mutual information
    s=='mir' ~ log(A*(A+B+C+D)/((A+B)*(A+C)))/log((A+D)/A),
    # Mutual information - relaive
    s=='ts' ~ 1/sqrt(A)*(A-(A+B)*(A+C)/n) # t-score
  )}#Association score (n*p)
```

5.17. 平滑得点

次の表(Y)は横に等間隔で連続する系列で（たとえば時系列），数値 D を記録した例です。このように数値の上下振動がある推移を移動しながら平均化した得点を「平滑得点」(Smooth Score)と呼びます。そのために「反復移動平均」(Recursive Moving Average: RMA)と呼ぶ方法を使います[15]。「移

[15] 佐竹元一郎（編）(2004)『経済の統計的分析』（中央経済社）p. 74-91

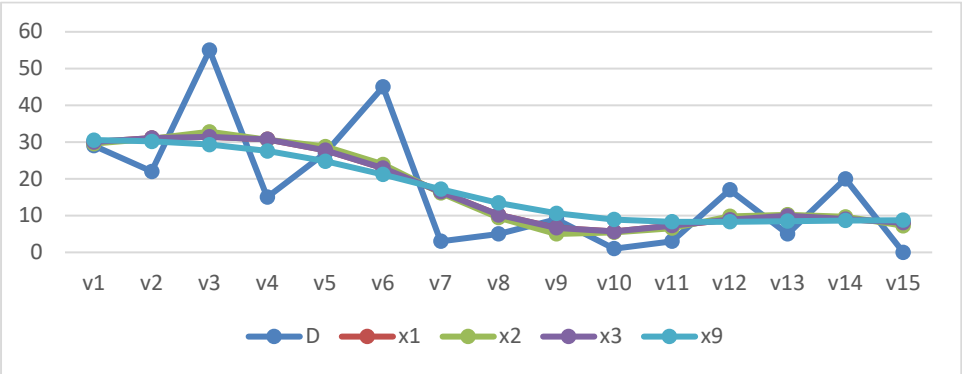
動平均値」(moving average)は，それぞれの数値とその前と後ろにそれぞれ k 個の数値の平均値ですが，反復移動平均では k を 1 に設定して， 直前・直後に連続する両側の数値の平均値を使います。これは「3 項平均値」(three-term average)と呼ばれます。v1 は直前にデータがないので，直前データとして該当データを代入して平均値を計算します。最後の v15 は直後にデータがないので，直後のデータとして該当データを使って平均値を計算します。たとえば RMA.1 の最初の数値 26.7 は直前の値がないので(29+29+22)/3 で求めます。次の 35.3 は直前と直後の値を使って(29+22+55)/3 です。その次は(22+55+15)/3 = 30.7 になります。以下同様に 3 つの数値の平均を計算し，右端の 6.7 は当該データを繰り返した 3 つの数値の平均で求めます（(20+0+0)/3）。

このように元の数値(D)にある振動が，RMA.1 の 3 項平均値によって「ならされて」平滑化します。しかし，それでも RMA.1 の各項にはまだ少し振動が残っているようです。そこで次の RMA.2 では，今求めた RMA.1 について同じ計算をします。たとえば，(26.7+26.7+35.3)/3 = 26.7; (26.7+35.3+30.7)/3=30.9。そして，RMA.3 は RMA.2 について同じ計算をします。以下同様の反復計算によって次第に数値の並びに振動が減り平準化の程度が高くなります。x

```
X1=Smooth(X,1); X2=Smooth(X,2); X3=Smooth(X,3); X9=Smooth(X,9) #平滑得点
Xp=rbind(D=X,x1=X1,x2=X2[1,],x3=X3[1,],x9=X9[1,]); R(Xp,1)
     v1   v2   v3   v4   v5   v6   v7   v8   v9 v10 v11  v12  v13  v14 v15
D  29.0 22.0 55.0 15.0 27.0 45.0  3.0  5.0  9.0 1.0 3.0 17.0  5.0 20.0  .0
x1 26.7 35.3 30.7 32.3 29.0 25.0 17.7  5.7  5.0 4.3 7.0  8.3 14.0  8.3 6.7
x2 29.6 30.9 32.8 30.7 28.8 23.9 16.1  9.4  5.0 5.4 6.6  9.8 10.2  9.7 7.2
x3 30.0 31.1 31.4 30.7 27.8 22.9 16.5 10.2  6.6 5.7 7.3  8.9  9.9  9.0 8.0
x9 30.5 30.2 29.3 27.6 24.8 21.2 17.2 13.4 10.6 8.9 8.3  8.3  8.5  8.6 8.7
```

次のグラフを見ると，データ(D)の振動が移動平均値を 1 回計算した RMA.1 でかなり平滑化されています。さらに反復計算を 2 回実行した RMA.2 では平滑化の効果が顕著です。RMA.3 は RMA.2 とほとんど差がありません。それでも平滑化を続けることは可能ですが，あまり続けすぎると平滑線は単調な線になります。平滑曲線はデータの平均的な推移を示すので，平均であることと推移を示すこと，という背反する性質を共にバランスよく示すようにしなければなりません。単調な平滑線は平均であることだけが強調されて推移の情報が犠牲にされています。

を参照しました。



移動平均はデータの両側の数値を含んだ平均値を使うので，反復数が少ないときデータの値と逆向きになることがあります。このグラフでは赤線で示した RCA.1 の v2 が本来ならば下向きになるはずですが上向きになり，逆に上向きのはずの v3 がやや下向きになっています。この反転現象はデータの解釈をミスリードする可能性があります(たとえば「v2 の時点でデータの数値が上がる」というような解釈)。

反復移動平均は個々のデータの変動よりも，むしろ全体的な推移の傾向を見るために使います。とくに連続した時系列データの分析のために有用です。そして，歴史資料などを扱う際，完全な資料がないため特定の年代のデータに欠損値があったり，大きな外れ値があったりするときに移動平均値が役立ちます。

次はデータ行列を行を平滑化(1 回：5 回)をした結果です。

```
BIND(D,R(Smooth(D,1),1),R(Smooth(D,5),1))
   c1 c2 c3 c4 c5 :   c1   c2   c3   c4   c5 :   c1   c2   c3   c4   c5
r1 10 19 14  7 12 : 13.0 14.3 13.3 11.0 10.3 : 13.3 13.0 12.5 11.8 11.3
r2 11  7 10  0  1 :  9.7  9.3  5.7  3.7  0.7 :  8.4  7.4  5.8  4.2  3.1
r3  0  0  1 12  1 :  0.0  0.3  4.3  4.7  4.7 :  1.2  1.9  2.9  3.8  4.3
r4  0  1  2  3  3 :  0.3  1.0  2.0  2.7  3.0 :  1.0  1.3  1.8  2.3  2.6
```

次はデータ行列を列を平滑化(1 回：5 回)をした結果です。

```
BIND(D,R(Smooth(D,1,2),1),R(Smooth(D,5,2),1))
   c1 c2 c3 c4 c5 :   c1   c2   c3  c4  c5 :  c1  c2  c3  c4  c5
r1 10 19 14  7 12 : 10.3 15.0 12.7 4.7 8.3 : 7.3 9.8 9.0 5.4 5.6
r2 11  7 10  0  1 :  7.0  8.7  8.3 6.3 4.7 : 6.1 8.0 7.7 5.4 4.8
r3  0  0  1 12  1 :  3.7  2.7  4.3 5.0 1.7 : 4.4 5.5 5.8 5.6 3.7
r4  0  1  2  3  3 :  0.0  0.7  1.7 6.0 2.3 : 3.2 3.8 4.5 5.6 3.0
```

ユーザー関数:

```
Smooth=function(X,n=1,s=1){
  if(s==2) X=t(X); nr=nrow(X); nc=ncol(X); W=X
  for(h in 1:n){
    X=W; for(i in 1:nr){for(j in 1:nc){
      if(j==1)        W[i,j]=(X[i,1]+X[i,j]+X[i,j+1])/3
```

```
        else if(j==nc) W[i,j]=(X[i,j-1]+X[i,j]+X[i,nc])/3
        else           W[i,j]=(X[i,j-1]+X[i,j]+X[i,j+1])/3
    }}}; if(s==2) t(W) else W
} #平滑化得点  n:回数, s=1:行, 2:縦
```

●移動平均

芝他(1984:9)は「時系列データから偶然誤差による変動を除き，時間 t_j (j=1, …, n)に関する滑らかなトレンドを得るための方法」として，次の「移動平均」(moving average: m_i)の式を提示しています。

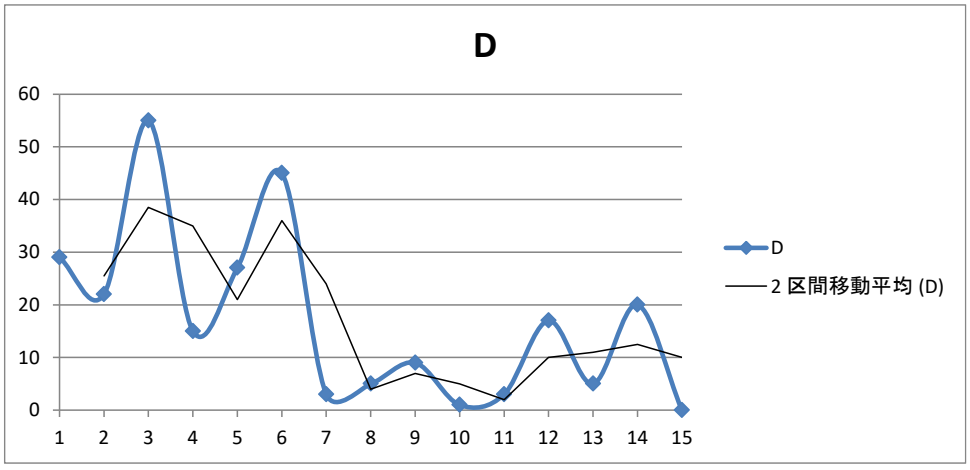
$$m_i = \frac{1}{2k+1} \sum_{i-k}^{i+k} x_j \quad (i = k+1, k+2, \ldots, n - k)$$

ここで，k は 1, 2, …, n/2 の中の任意の整数，n はデータ数です。よって，それぞれの i について，i-k から i+k までの和を 2k+1 で割って平均値を求めます(和を 2k ではなく 2k+1 で割る理由は x_j の値も含めているからです)。

なお，Excel で計算する移動平均 $m_i(Excel)$は次の式のように x_j と先行する k 個のデータの平均値を使っています。

$$m.Excel_{\,i} = \frac{1}{k+1} \sum_{i-k}^{i} x_j \quad (i = k+1, k+2, \ldots, n - k)$$

次は Excel の「2 区間移動平均」のグラフです[16]。移動平均値なので x 軸の 1 は計算から除外され，グラフ線は x 軸の 2 から開始します。



このように，芝他(1984:9)の移動平均ではデータ全体の両端の値がなく，

[16] グラフで「散布図」→「平滑線とマーカー」を選択し，平滑線を選択し右クリック「近似曲線の追加」から「移動平均」を選択し，「区間」(k)を 2 とします。

Excel の移動平均では左端の値がないため，反復計算には向いていません[17]。また，とくに x 軸の値が少ないデータを扱うとき両端または左端が欠けていると，その解釈ができなくなり，大きな情報を失います[18]。そこで反復移動平均法では両端または左端であってもそこで使えるデータを考慮する，という方針で平均値の計算に組み込みました。

5.18. 度数分布得点

実測値を一定の大きさの階級に分けて，階級ごとの度数を計算したものを「度数分布得点」(Frequency distribution score)と呼びます。次のデータ(X)を使います。

X	v1	v2	v3
d1	45	48	66
d2	56	59	54
d3	58	51	78
d4	77	72	20
d5	43	44	32
d6	58	34	90
d7	50	53	100

はじめに，データ X の第 1 行を取り出して，その度数分布を調べます。

```
 K=X[1,]; K #1 行
   v1 v2 v3
d1 45 48 66
```

データ(45, 48, 66)の分布を間隔=10 で見ると,次のようになります。

```
FreqDist(K,10) #度数分布.間隔=10
  value count sum
1    0-     0   0
2   10-     0   0
3   20-     0   0
4   30-     0   0
5   40-     2  93
6   50-     0   0
7   60-     1  66
```

このように，40-のグループに 2 個，60-のグループに 1 個データが見つかります。それぞれのグループの和は 93, 66 です。次に，1 行だけでなく，

[17] 反復のたびに両端または左端の値が失われてデータの規模が小さくなります。
[18] とくに区間(k)の値を大きくすると欠損部が拡大していきます。

データ行列全体の度数分布を見ると，次のようになります。

```
> Bind(X,FDS(X,10,1)) #度数分布得点: 行
   v1 v2  v3 : 0- 10- 20- 30- 40- 50- 60- 70- 80- 90- 100-
d1 45 48  66 :  .   .   .   .   2   .   1   .   .   .    .
d2 56 59  54 :  .   .   .   .   .   3   .   .   .   .    .
d3 58 51  78 :  .   .   .   .   .   2   .   1   .   .    .
d4 77 72  20 :  .   .   1   .   .   .   .   2   .   .    .
d5 43 44  32 :  .   .   .   1   2   .   .   .   .   .    .
d6 58 34  90 :  .   .   .   1   .   1   .   .   .   1    .
d7 50 53 100 :  .   .   .   .   .   2   .   .   .   .    1
```

たとえば，d1 行{45, 48, 66}の成分は，それぞれ 40, 40, 60 の階級に入るので，40 の階級の度数が 2 になり，60 の階級の度数が 1 になります。上右表全体を見ると，どの階級で比較的頻度が高いのかを観察することができます。

次に，同じ関数のパラメータ cs を 2 として，それぞれの個数ではなく，度数の和を出力させました。

```
Bind(X,FDS(X,10,1,cs=2)) #度数分布得点: 行, cs=2:sum
   v1 v2  v3 : 0- 10- 20- 30- 40- 50- 60- 70- 80- 90- 100-
d1 45 48  66 :  .   .   .   .  93   .  66   .   .   .    .
d2 56 59  54 :  .   .   .   .   . 169   .   .   .   .    .
d3 58 51  78 :  .   .   .   .   . 109   .  78   .   .    .
d4 77 72  20 :  .   .  20   .   .   .   . 149   .   .    .
d5 43 44  32 :  .   .   .  32  87   .   .   .   .   .    .
d6 58 34  90 :  .   .   .  34   .  58   .   .   .  90    .
d7 50 53 100 :  .   .   .   .   . 103   .   .   .   .  100
```

次は同じデータの列による度数分布得点です。個数と和を左右に並べました。今度は v1, v2, v3 のそれぞれの列の階級の分布の様子がわかります。

```
Bind(FDS(X,10,2,cs=1), FDS(X,10,2,cs=2)) #度数分布得点: 列,
cs=1:count, 2:sum
     v1 v2 v3 :  v1  v2  v3
  0-  .  .  . :   .   .   .
 10-  .  .  . :   .   .   .
 20-  .  .  1 :   .   .  20
 30-  .  1  1 :   .  34  32
 40-  2  2  . :  88  92   .
 50-  4  3  1 : 222 163  54
 60-  .  .  1 :   .   .  66
 70-  1  1  1 :  77  72  78
 80-  .  .  . :   .   .   .
 90-  .  .  1 :   .   .  90
100-  .  .  1 :   .   . 100
```

一般に度数分布表は度数を適当な間隔で分類しますが，以下では度数の順位のグループ(rank)で分類します。はじめに X の第 1 列の度数を見ます。

```
K=X[,1]; K #1列
[1] 45 56 58 77 43 58 50
> FreqDistR(K,3) #度数分布.順位, 分割=3
  rank value count sum
1  1-2 43-45     2  88
2  3-4 50-56     2 106
3  5-7 58-77     3 193
```

次に，データ行列 X 全体の列の度数を見ます。

```
FDSr(X,3,2) #度数分布得点.順位
     v1  v2  v3
1-2  88  78  52
3-4 106  99 120
5-7 193 184 268
> X
   v1 v2  v3
d1 45 48  66
d2 56 59  54
d3 58 51  78
d4 77 72  20
d5 43 44  32
d6 58 34  90
d7 50 53 100
```

ユーザー関数:

```
FreqDist=function(A,it=10,s=0){A=unlist(A)
  f=ifelse(s==0,floor,Rnd); Id=f(A/it)+1; mx=max(Id); Ct=Sm=rep(0,mx)
  for(i in 1:length(Id)) {id=Id[i]; Ct[id]=Ct[id]+1; Sm[id]=Sm[id]+A[i]}
  st=ifelse(s==0,'','-'); data.frame(value=st&0:(mx-1)*it&'-', count=Ct,
sum=Sm)
} #度数分布 Frequency distribution (A:array, it: interval,s=0:floor, 1:Rnd)
```

```
FDS=function(X,it,s=1,fr=0){
  #s=1:row, 2:col, it: interval,fr=0:floor, 1:Rnd
  if(s==2) X=t(X); L=apply(X,1,function(X){FreqDist(X,it,fr)})
  mx=max(sapply(L, nrow)); W=matrix(0,length(L),mx); Cn=NULL
  for(i in 1:length(L)){for(j in 1:nrow(L[[i]])){
    Cn[j]=L[[i]][j,1]; W[i,j]=L[[i]][j,2]
  }}; W[is.na(W)]=0; rownames(W)=rownames(X); colnames(W)=Cn
  if(s==2) {W=t(W); W=JustRrownames(W)}; W=ifelse(W==0,'.',W); JustR(W)
} #度数分布得点 Frequency distribution score
```

```
FreqDistR=function(A,d=3){ #A: array, d: number of division
  A=unlist(A); A=sort(A); dv=Rnd(length(A)/d); Fr=To=Ct=Sm=Vf=Vt=NULL
  for(i in 1:d){
    Fr[i]=(i-1)*dv+1; To[i]=ifelse(i<d,i*dv,length(A))
```

```
    Ct[i]=length(A[Fr[i]:To[i]]); Sm[i]=sum(A[Fr[i]:To[i]])
    Vf[i]=A[Fr[i]]; Vt[i]=A[To[i]]
  }; data.frame(rank=Fr&'-'&To, value=Vf&'-'&Vt, count=Ct, sum=Sm)
} #Frequency distribution by rank
```

```
FDSr=function(X,d,s=1){ #s=1:row, 2:col, d: division
  if(s==2) X=t(X); L=apply(X,1,function(X){FreqDistR(X,d)})
  mx=max(sapply(L, nrow)); W=matrix(0,length(L),mx); Cn=NULL
  for(i in 1:length(L)){for(j in 1:nrow(L[[i]])){
    Cn[j]=L[[i]][j,1]; W[i,j]=L[[i]][j,4]
  }}; W[is.na(W)]=0; rownames(W)=rownames(X); colnames(W)=Cn
  if(s==2)  {W=t(W);  W=JustRrownames(W)};  W=ifelse(W==0,'.',W);
JustR(W)
} #度数分布得点:  Frequency distribution score  by rank
```

●相関表

最後に見るのは次の両軸による階級得点です。

```
CorTable(X,10,5,0) #相関表 Correlation table
 *row:v1 / column:v2
    0- 5- 10- 15- 20- 25- 30- 35- 40- 45- 50- 55- 60- 65- 70-
 0-  .  .   .   .   .   .   .   .   .   .   .   .   .   .   .
10-  .  .   .   .   .   .   .   .   .   .   .   .   .   .   .
20-  .  .   .   .   .   .   .   .   .   .   .   .   .   .   .
30-  .  .   .   .   .   .   .   .   .   .   .   .   .   .   .
40-  .  .   .   .   .   .   .   .   1   1   .   .   .   .   .
50-  .  .   .   .   .   .   1   .   .   .   2   1   .   .   .
60-  .  .   .   .   .   .   .   .   .   .   .   .   .   .   .
70-  .  .   .   .   .   .   .   .   .   .   .   .   .   .   1
```

このように，縦軸の v1 と v2 の階級を縦と横の両軸に置いて，それぞれの階級がクロスする位置に該当する度数を入れます。この表は「相関表」(correlation table)と呼ばれます。次のユーザー関数 CorTable は入力データの第 1 列と第 2 列を対象にし，それぞれの間隔を指定し(i1, i2)，s で間隔を「切り捨て」(s=0:floor)，または「四捨五入」(s=1:Rnd)で設定します。間隔を切り捨てで設定すると，グループの数値は開始点を示し，四捨五入で設定すると，グループの数値は間隔の中央を示します。

ユーザー関数:

```
CorTable=function(X,i1=10,i2=10,s=0){
  # X: df,mat(col.1=>row, col.2=>col), i1,i2:interval, s=0:floor, 1:Rnd
  f=ifelse(s==0,floor,Rnd); C1=f(X[,1]/i1)+1; C2=f(X[,2]/i2)+1
  m1=max(C1); m2=max(C2); M=matrix(0,m1,m2)
```

```
  for(i in 1:length(C1)){x=C1[i]; y=C2[i]; M[x,y]=M[x,y]+1}
  t=ifelse(s==0,'','-')
  rownames(M)=t&(0:(m1-1))*i1&'-'; colnames(M)=t&(0:(m2-1))*i2&'-'
  tit=' *row:'&colnames(X)[1]&' / column:'&colnames(X)[2]&'¥n'
  M=JustRrownames(M); M=ifelse(M==0,'.',M); cat(tit); JustR(M)
} #相関表 Correlation table
```

5.19. 欠損値補完得点

言語の地理・歴史資料などを分析するとき，特定の地域・年代で資料(数値)が見つからないことがあります。R ではそのような数値を NA (not available)として扱います。その NA については，(1) この資料の分析を断念する，(2) 資料がない特定の数値 NA としてそのまま分析する(後述)，(3) NA のある行・列を除去する(後述)，(4) 適当な方法で欠損値を補完する，という選択肢が考えられます[19]。それぞれの選択肢には長所と短所がありますが，ここでは (4) 欠損値を補完する簡単な方法を説明します。

(1) 平均値・中央値による欠損値補完

データ行列の全体の傾向を見ることを目的とするときに，欠損値にデータの平均値または中央値が補完されることがあります。平均値または中央値を補完する理由は欠損値をゼロとするよりも全体として大きな変化・影響が生まれないためです。

次は，サンプルデータ D の一部を NA と見做した例です。

```
X=D; X[1,3]=NA; X[2,1]=NA; X[3,3]=NA; X[4,5]=NA; X=AsNum(X); X
#サンプルデータ
   c1 c2 c3 c4 c5
r1 10 19 NA  7 12
r2 NA  7 10  0  1
r3  0  0 NA 12  1
r4  0  1  2  3 NA
```

このデータを昇順で並べ替えると，分布がかなり偏っていることがわかります。

```
> K=sort(c(X)); K; FreqDist(K,5); FreqDistR(K,5)
 [1]  0  0  0  0  1  1  1  2  3  7  7 10 10 12 12 19
```

そのことは度数分布を見るとさらにはっきりとします。

[19] しばしば欠損値をゼロにした分析を見ることがありますが，これは結論に大きなバイアスをつけることになるので避けるべきです。

```
  value count sum
1    0-     9   8
2    5-     2  14
3   10-     4  44
4   15-     1  19

   rank value count sum
1   1-3   0-0     3   0
2   4-6   0-1     3   2
3   7-9   1-3     3   6
4 10-12  7-10     3  24
5 13-16 10-19     4  53
```

このように低頻度のデータ数が多く(long tail)，顕著に高頻度のデータ数が限られていることは言語要素の出現頻度調査でしばしば観察されます。そのときは平均値よりも中央値を使用すべきです。平均値を使わない理由は，平均値(=5.3)が最大値(=19)やその近傍の値の影響を大きく受けて最大値に近づいているためです。このような平均値をデータを代表する値(普通の値)と見なすことができません。平均値(x:5.3)と中央値(y:2.5)の乖離の程度は両者の対照値(x-y)/(x+y)=0.36 を見るとわかります。対照値の範囲は[-1, 1]なので，0.36 は非常に大きな数値です。

```
mean(X,na.rm=T); median(X,na.rm=T); MeanMedianContrast(X)
[1] 5.3125
[1] 2.5
[1] 0.36
```

そこで，欠損値をデータの中央値で補完すると次のようになります。最初はそれぞれの行の中央値で NA を補完しています(NA=11)。次は列の中央値，最後は全体の中央値で補完しています。行・列・全の選択は，分析の目的に従います。

```
Bind(Ar(X,median,0,1),Impute(X,median,1))
#中央値による補完: 行
   c1 c2 c3 c4 c5 median : c1 c2  c3 c4  c5
r1 10 19 NA  7 12   11.0 : 10 19  11  7  12
r2 NA  7 10  0  1    4.0 :  4  7  10  0   1
r3  0  0 NA 12  1    0.5 :  0  0 0.5 12   1
r4  0  1  2  3 NA    1.5 :  0  1   2  3 1.5

> Bind(Ac(X,median,0,1),Impute(X,median,2))
#中央値による補完: 列
        c1  c2  c3  c4  c5 : c1 c2 c3 c4 c5
r1      10  19  NA   7  12 : 10 19  6  7 12
r2      NA   7  10   0   1 :  0  7 10  0  1
r3       0   0  NA  12   1 :  0  0  6 12  1
r4       0   1   2   3  NA :  0  1  2  3  1
median  .0 4.0 6.0 5.0 1.0 :
```

```
> Bind(Aa(X,median,0,1),Impute(X,median,3))
#中央値による補完: 全
       c1 c2 c3 c4 c5 median :  c1 c2  c3 c4  c5
r1     10 19 NA  7 12      - :  10 19 2.5  7  12
r2     NA  7 10  0  1      - : 2.5  7  10  0   1
r3      0  0 NA 12  1      - :   0  0 2.5 12   1
r4      0  1  2  3 NA      - :   0  1   2  3 2.5
median  -  -  -  -  -    2.5 :
```

ユーザー関数:

```
MeanMedianContrast=function(A){
  me=mean(A,na.rm=T); md=median(A,na.rm=T); (me-md)/(me+md)
} #平均値:中央値の対照値 Mean median contrast
```

```
Impute=function(X,f=mean,s=1){ #f=mean, median, s=1:行,2:列,3:全
  #X=AsNum(X);
  W=X; Id=which(is.na(X), arr.ind=T)
  if(s==1|s==2) {
    for(i in 1:nrow(Id)){
      if(s==1) W[Id[i,1],Id[i,2]]=f(X[Id[i,1],], na.rm=T)
      if(s==2) W[Id[i,1],Id[i,2]]=f(X[,Id[i,2]], na.rm=T)
    }
  }else W[Id]=median(X, na.rm=T); W
} #欠損値補完 Imputation of missing value by mean/median
```

(2) 隣接値による欠損値補完

データ行列に欠測値があるとき，隣接の値を参照することが可能であると判断したときに使用します。たとえば，行がそれぞれ連続する年代を示すときは，NA の前後の年代の平均を補完します。列が地域の連続性のある地理的配置を示す場合は，NA にその上下の数値の平均を補完します。行・列ともに連続性があれば，上下・左右の数値 4 個の平均を補完します。NA が行列の端にあるときや隣接値が NA であるときは，その数値を含めない平均値を使います。

```
Bind(X,R(ImputeN(X,1),1)) #隣接値による補完: 行
   c1 c2 c3 c4 c5 :   c1   c2   c3   c4   c5
r1 10 19 NA  7 12 : 10.0 19.0 13.0  7.0 12.0
r2 NA  7 10  0  1 :  7.0  7.0 10.0  0.0  1.0
r3  0  0 NA 12  1 :  0.0  0.0  6.0 12.0  1.0
r4  0  1  2  3 NA :  0.0  1.0  2.0  3.0  3.0
```

```
> Bind(X,R(ImputeN(X,2),1)) #隣接値による補完: 列
   c1 c2 c3 c4 c5 :   c1   c2   c3   c4   c5
r1 10 19 NA  7 12 : 10.0 19.0 10.0  7.0 12.0
r2 NA  7 10  0  1 :  5.0  7.0 10.0  0.0  1.0
r3  0  0 NA 12  1 :  0.0  0.0  6.0 12.0  1.0
r4  0  1  2  3 NA :  0.0  1.0  2.0  3.0  1.0

Bind(X,R(ImputeN(X,3),1)) #隣接値による補完: 全
   c1 c2 c3 c4 c5 :   c1   c2   c3   c4   c5
r1 10 19 NA  7 12 : 10.0 19.0 12.0  7.0 12.0
r2 NA  7 10  0  1 :  5.7  7.0 10.0  0.0  1.0
r3  0  0 NA 12  1 :  0.0  0.0  6.0 12.0  1.0
r4  0  1  2  3 NA :  0.0  1.0  2.0  3.0  2.0
```

●欠損値のある行・列の除去

多くの欠損値がある行・列だけを除去することがあります。たとえば，次のサンプルデータでは，r1, r2 行, c1, c3 列に NA があります。

```
X=D; X[1,3]=NA; X[2,1]=NA; X=AsNum(X); X #サンプルデータ
   c1 c2 c3 c4 c5
r1 10 19 NA  7 12
r2 NA  7 10  0  1
r3  0  0  1 12  1
r4  0  1  2  3  3
```

はじめに NA を含む行を除去し，次に NA を含む列を除去します。

```
> DelNA(X,1) #NA を含む行を除去
   c1 c2 c3 c4 c5
r3  0  0  1 12  1
r4  0  1  2  3  3

> DelNA(X,2) #NA を含む列を除去
   c2 c4 c5
r1 19  7 12
r2  7  0  1
r3  0 12  1
r4  1  3  3
```

なお，次のように，各種の R 関数は na.rm を指定すれば NA を除いて計算した数値を返します。

```
> sum(X, na.rm=T)
[1] 89
```

ユーザー関数:

```
DelNA=function(X,s=1){
  if(s==1) X[which(rowSums(is.na(X))==0),]
  else X[,which(colSums(is.na(X))==0)]
```

```
} #NA を含む行・列を除去 (s=1:行, 2:列)
```

5.20. 対数得点

言語単位の出現頻度には非常に大きな差異が観察されることがあります。出現頻度に桁違いに大きな差異があるときには，対数の底を 10 にして，その桁数を比較することが有効です。また，桁の違いほど大きくなければ，対数の底を 2 にすることも考えられます。R では次の関数が用意されています。

```
log2(8); log(8,2) #binary logarithm …(1)
[1] 3
[1] 3
log10(10000); log(10000,10) #common logarithm …(2)
[1] 4
[1] 4
log(0); log1p(0) #logarithm of 1 plus …(3)
[1] -Inf
[1] 0
log(0+1)/log(10) #logarithm of 1 plus with base:10 …(4)
[1] 0
```

上の(1)の R 関数 log2 は真数(true value)=8，底(base)=2 のバイナリー対数(binary logarithm)を返します。(2)の log10 は底=10 の常用対数(common logarithm)を返します。(3)の log1p は真数に 1 を加えた自然対数(natural logarithm of 1 plus)を返します。log1p は自然対数なので底を指定できません。底を指定するときは，(4)のように自然対数の割り算にします。

次のように，ユーザー関数 Log1 はデータ(真数)と底を指定して真数＋1 の対数を返します。

```
Bind(D,R(Log1(D,2),1)) #真数＋1 の対数
   c1 c2 c3 c4 c5 :  c1  c2  c3  c4  c5
r1 10 19 14  7 12 : 3.5 4.3 3.9 3.0 3.7
r2 11  7 10  0  1 : 3.6 3.0 3.5  .0 1.0
r3  0  0  1 12  1 :  .0  .0 1.0 3.7 1.0
r4  0  1  2  3  3 :  .0 1.0 1.6 2.0 2.0
```

上の出力を見ると，真数が 0 のとき，対数は-Inf ではなく，ゼロ(0)が出力されているので,出力データの和や平均値などを求めることができます。

ユーザー関数:

```
Log1=function(x, b=10) log(x+1)/log(b) #logarithm of 1 plus with base:b
```

■スペイン語の文字の頻度

次はスペイン語テキスト中の文字の頻度を対数（底：2）に変換した表で

す。左表にはゼロ，2, 3 などという非常に少ない頻度の文字と e, a, o などの高頻度の文字(3068, 2928, 2060)がありますが，それらを対数変換をすると比較しやすい数値に変わりました。

Form	1_Madrid
a	2925
b	326
c	834
d	978
e	3068
f	117
g	298
h	299
i	1143
j	124
k	3
l	1153
m	748
n	1438
o	2060
p	641
q	335
r	1452
s	1853
t	1053
u	989
v	301
w	0
x	17
y	367
z	71
á	125
é	152
í	217
ñ	57
ó	112
ú	35
ü	2

Log(2)	1_Madrid
a	11.5
b	8.3
c	9.7
d	9.9
e	11.6
f	6.9
g	8.2
h	8.2
i	10.2
j	7.0
k	1.6
l	10.2
m	9.5
n	10.5
o	11.0
p	9.3
q	8.4
r	10.5
s	10.9
t	10.0
u	9.9
v	8.2
w	.0
x	4.1
y	8.5
z	6.1
á	7.0
é	7.2
í	7.8
ñ	5.8
ó	6.8
ú	5.1
ü	1.0

5.21. 行列接合得点

次の出力はデータ行列 D と，D の行と列を接合した結果を示します。たとえば，接合した行列の第 1 行にはデータ行列の r1:c1=10 が r1 と c1 のそれぞれの列に代入されています。このようにして，データ行列のすべての要素を該当する位置に代入した行列を「行列接合得点」(Row and column joint score)と呼びます。

```
X=RowColJoint(D); Bind(D, X)
   c1 c2 c3 c4 c5 : r1 r2 r3 r4 c1 c2 c3 c4 c5
r1 10 19 14  7 12 : 10  0  0  0 10  0  0  0  0
r2 11  7 10  0  1 : 19  0  0  0  0 19  0  0  0
r3  0  0  1 12  1 : 14  0  0  0  0  0 14  0  0
r4  0  1  2  3  3 :  7  0  0  0  0  0  0  7  0
                  : 12  0  0  0  0  0  0  0 12
                  :  0 11  0  0 11  0  0  0  0
                  :  0  7  0  0  0  7  0  0  0
                  :  0 10  0  0  0  0 10  0  0
                  :  0  0  0  0  0  0  0  0  0
                  :  0  1  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  1  0  0  0  1  0  0
                  :  0  0 12  0  0  0  0 12  0
                  :  0  0  1  0  0  0  0  0  1
                  :  0  0  0  0  0  0  0  0  0
                  :  0  0  0  1  0  1  0  0  0
                  :  0  0  0  2  0  0  2  0  0
                  :  0  0  0  3  0  0  0  3  0
                  :  0  0  0  3  0  0  0  0  3
```

行列接合得点そのものは入力行列と同じ情報を有しますが，かえって入力行列よりも無駄が多く，解釈が困難になります。しかし，行列接合得点に各種の統計処理を行うことによって，行の情報と列の情報を同時に扱うことが可能になります。たとえば，次の出力はデータ行列 D の行と列のそれぞれの相関行列を示します。

```
X=R(cor(t(D)),3); Y=R(cor(D),3); Bind(X,Y)
       r1     r2     r3     r4 :     c1     c2     c3     c4    c5
r1  1.000  0.387 -0.683 -0.323 :  1.000  0.787  0.944 -0.480  .436
r2  0.387  1.000 -0.670 -0.840 :  0.787  1.000  0.945 -0.092  .896
r3 -0.683 -0.670  1.000  0.586 :  0.944  0.945  1.000 -0.331  .709
r4 -0.323 -0.840  0.586  1.000 : -0.480 -0.092 -0.331  1.000  .140
                               :  0.436  0.896  0.709  0.140 1.000
```

一方，次の行列接合得点の相関行列では，行と列の間に見られる相関を見ることができます。

```
R(cor(X),3)
       r1     r2     r3     r4     c1     c2     c3     c4     c5
r1  1.000 -0.232 -0.145 -0.250  0.096  0.559  0.269 -0.056  0.300
r2 -0.232  1.000 -0.115 -0.198  0.424  0.034  0.247 -0.158 -0.132
r3 -0.145 -0.115  1.000 -0.124 -0.089 -0.084 -0.094  0.826 -0.079
r4 -0.250 -0.198 -0.124  1.000 -0.154 -0.132 -0.115 -0.016  0.026
c1  0.096  0.424 -0.089 -0.154  1.000 -0.104 -0.124 -0.123 -0.107
c2  0.559  0.034 -0.084 -0.132 -0.104  1.000 -0.116 -0.115 -0.100
c3  0.269  0.247 -0.094 -0.115 -0.124 -0.116  1.000 -0.137 -0.119
c4 -0.056 -0.158  0.826 -0.016 -0.123 -0.115 -0.137  1.000 -0.118
c5  0.300 -0.132 -0.079  0.026 -0.107 -0.100 -0.119 -0.118  1.000
```

上の枠で囲んだ部分は行 r1:4 と列 c1:5 の相関を示します。その最大値は r3:c4 の.826 ですが，データ行列 r3:c4 はデータの最大値ではありません。相関係数は個別の頻度の関係ではなく，2 つの変数にあるすべての頻度の関係を示すからです。そして，行列接合得点の相関行列の変数はデータ行列の行と列すべてを同時に含んでいます。データ行列の行 r1:4 の間と，列 c1:5 の間の共起回数はそれぞれゼロであり，その相関係数はすべて負になります。

```
RowColJoint=function(X){
  nr=nrow(X); nc=ncol(X); W=matrix(0,nr*nc,nr+nc); n=0
  rownames(W)=1:(nr*nc); colnames(W)=c(rownames(X),colnames(X))
  for(i in 1:nr){for(j in 1:nc){n=n+1; W[n,i]=W[n,nr+j]=X[i,j]}}; W
} #行列接合得点  Row and column joint score
```

(終)