April 23, 2008 V1R3 A
April 28, 2010 V2R0

情報リテラシー演習

S 言語と R 言語 第二回

離散事象のための確率モデル (二項分布)

w.yamamoto

1 関数プログラミング

S 言語 (および R 言語) で関数を定義するには、function() を用いる。

```
function(引数) return(返り値);
function(引数) {
   関数の実行文;

   return(返り値);
}
```

引数は複数指定することができるし、省略することもできる。関数の返り値は一つのオブジェクトしか渡せないが、list オブジェクトやデータフレームを返すことで、複雑なデータを渡すことが可能となっている。

以下ではまず、一つの整数を与えられて、一つの実数オブジェクトを返す関数の例として、Fibonacci 数列の第 n 項を計算する関数 fibonacci() を与える。

```
> fibonacci <- function(n) {
+   if (n<=0) {
+     return(0)
+   } else if(n<=2) {
+     return(1)
+   } else {
+     x <- 1;  y <- 1;
+     for ( i in c(3:n) ) {
+        z <- x + y
+        x <- y
+        y <- z
+     }
+     return(z)
+   }
+ }
```

これを第 n 項まで返す関数に改造するには、返り値を実数ベクトルとする必要がある。そのことは、下記のように書ける。

```
> fibonacci.list <- function(n) {
+   if (n<=0) {
+     return(NA)
+   } else if(n==1) {
+     return(c(1))
+   } else if(n==2) {
+     return(c(1,1))
+   } else {
+     x <- 1;  y <- 1;
+     fib <- c(1,1)
+     for ( i in c(3:n) ) {
+        z <- x + y
+        fib <- append(fib,z)
+        x <- y
+        y <- z
+     }
+     return(fib)
+   }
+ }
```

更に複雑な返り値も与えられる例として、第 n 項までの任意の項を返す関数に改造すると次のようになる。

```
> fibonacci.sublist <- function(n.sublist) {
+   n <- max(n.sublist)
+   if (n<=0) {
+     return(NA)
+   } else if(n==1) {
+     fib <- c(1)
+   } else if(n==2) {
+     fib <- c(1,1)
+   } else {
+     x <- 1;  y <- 1;
+     fib <- c(1,1)
+     for ( i in c(3:n) ) {
+        z <- x + y
+        fib <- append(fib,z)
+        x <- y
+        y <- z
+     }
+   }
+   return(list(sublist=fib[n.sublist],n=n.sublist))
+ }
```

これらの関数の実行例は、次の通り。特に `fibonacci.sublist()` を用いて、第 100 項から第 1000 項まで、100 飛ばしで計算していることを、理解して欲しい。

```
> fibonacci(10)
> fibonacci.list(10)
> fibonacci.1000 <- fibonacci.sublist(c(1:10)*100)
> fibonacci.100$n.sublist
> fibonacci.100$n
> plot(fibonacci.100$n.sublist,fibonacci.100$n)
```

このように、関数を定義することから、構造化プログラミングは始まる。

2 確率シミュレーション

システム工学科では、2年次に「確率論」と「統計学」をそれぞれ、「確率統計学第一」と「確率統計学第二」として学ぶ機会がある。今回から数回は、確率論、についての課題である。理論については、講義での解説を待たねばならないが、離散確率分布としては一番簡単な二項分布について、数式ではなく、確率シミュレーションにより、性質を確認する課題を行う。

2.1 確率論の入り口

多くの確率論の講義では「繰り返し起こる無作為 (ランダム) な事象 (以下、確率事象)」に対して「互いに独立な繰り返し観測を行う」状況を考える。

対象や事象を観測・測定した結果を、確率論では 観測値 という。観測値の列にランダム性がみられず、一定のパターンを示す、あるいは決まった関係式に従って変化しているとき、方程式、あるいは漸化式などを用いて、将来の観測値まで完全に予測できる。

- 萩野先生は、月曜日から土曜日まで、朝 5 時に来て、夜 10 時に帰る。(確定事象)
- システム工学科事務室は平日は 9 時に開室し、5 時に閉室する。土曜日は誰もいない。(確定事象)
- 西五号館の教室はすべて、毎日、決められた時間に掃除していただいているが、システム工学科計算機室はその対象外なので、常に掃除されない。(確定事象)
- 太陽系の惑星は太陽を中心に、ケプラーの法則に従って、運動している。(天体の運動は予測可能、シミュレーション可能)
- 国立大学の教員は裁量労働制であり、毎月振り込まれる給料は人事院勧告によって決まる。税制度や保険制度の変更の影響を受ける以外は、毎月何時間働こうが、休職するか悪事を働かない限り、振り込まれる給料は変わらない。(確定事象)

これらは予測が可能なため、確率事象ではない。確定事象あるいは確定論的事象と呼ばれる。

確率事象は「無作為性」を持ち、完全に予測できてはならない。繰り返し観測にも「独立性」を保証する必要がある。「無作為性」と「独立性」が成り立たなければ、確率論に基づく計算 (確率計算) には狂いが生じるので、この二つは大事な概念である。

簡単に思いつく、例示できる、あるいは実際に経験したことのある、身近な 確率事象 には、次のようなものがある。

- コインを何回か投げ、各面が上になる割合を観測する。ただしコインや投げ方に如何様はないとする。(二項分布)
- 同じことを、賽子で行う。(多項分布)
- 道路の混雑状況を把握するために、ある交差点に進入し、出て行く自動車を、車種・時間ごとに調査する。ただし、五十日のみの調査ではないことを保証するために、１日だけでなく何日か、調査を実施する。(待ち行列理論)
- 先週の資料の p.8 に「の」の字がいくつあるかを数えてみる。(計数誤差)
- 基礎科学実験の「重力加速度の測定」で、振り子の周期の測定に、フラッシュ光を用いるピート法を用い、複数回測定して、平均値を用いる。(測定誤差)
- 基礎科学実験の「放射戦の計測」で、N_0 個の放射線原子が、ある時間に N 個崩壊するとすると、N はポアソン分布に従う、と考察しているはず。
- 麻雀を始める前に、牌は完全に攪拌してあり、積むときにも作為のない (積み込みのない) ように積まれていると仮定する。(確率計算の前提として、無作為性を保証)
- 期末試験の点数は、勉強した内容および努力と、出題された問題とで決まり、予め問題が流出していることはない。また、学生の試験の出来不出来はなく実力を測定できているか、出来不出来があっても成績評価に対しては無視できる程度である。(測定誤差)

以上はいずれも、無作為性のある事象である。また観測には独立性を仮定しており、実験テキストでも、暗黙裏にそのように扱われている。

観測値の列が、傾向はあっても一定のパターンを示すことはなく、その変化に作為が見られず、無作為 (ランダム) とみなせるときに、確率論はそれらの観測値が従う法則を数理モデル化する方法を与える。

これらの準備の下、確率変数が従う法則を確率分布と呼ぶ。この法則が観測するごとに変化せず、しかも毎回の観測の間にいかなる関連性もない (からそこをモデルに含める必要がない)、という意味が、「X_1,X_2,...,X_n は独立に同一の分布に従う」との宣言には込められている。

以上のことを少し抽象化し、「n 回の独立試行を行うとき、各試行の結果を確率変数 $X_1, X_2, \ldots, X_n$ で表す。$X_1, X_2, \ldots, X_n$ は互いに独立に、同一の分布 $F(x;\theta)$ に従うと仮定する。」という。そして、実際に試行を行って得た観測値を 実現値 と呼び、「$x_1, x_2, \ldots, x_n$ を確率変数 $X_1, X_2, \ldots, X_n$ の実現値とする」という。

2.2 乱数と疑似乱数

乱数とは、上で述べた確率変数の実現値と見なせる数、である。予測できてはいけないし、特定のパターンをとってもいけない。コンピュータが発明される前には、10 面体や正 20 面体のサイコロ (乱数賽)、数字をランダムに並べた乱数表、などが乱数の生成に用いられた。

一方、現在では、そのような数列を得るのに、様々な仕組みが考案されている。[1]

- 熱雑音 (抵抗内の電子の不規則な熱振動によって生じる雑音) を利用した物理乱数。
- トランジスタのナノ・スケール領域での揺らぎを利用した物理乱数。
- ラジオからの雑音の観測。

[1] これらは商品化されたり、インターネット経由で利用可能なサービスが提供されたりするなど、実用化が進んでいる。近年、疑似乱数は暗号化技術への応用が注目されており、一層の小型化、高速化、停電力化が進むと予想される。現在でも、USB モジュールとして商品化された製品が数万円で購入できるが、未だにコンピュータでシミュレーションを行うときに利用されるのは、疑似乱数である。

しかし、その二つの間の時代、コンピュータが発明されて暫くは、これらの技術は利用可能ではなかった。そのため、「乱数に見える数列」(以下、疑似乱数 と呼ぶ) を発明する試みがフォン・ノイマンを始めとして、多くの研究者によってなされた。疑似乱数の研究では、0 から $2^n - 1$ までの整数を取る頻度が一様になり、しかもどこから切り出しても、ランダムにしか見えないような値の数列をどのように生成するか、を目標とする。コンピュータのプログラムとして記述可能な数列である以上、予測不能性は保証されないが、それでも無作為かつ独立とみなせることが要求される。[2]

後述するように、他の確率分布に従う疑似乱数の生成は、整数値をとる一様な疑似乱数を変換することで得られる。

3 二項分布

表が上になる確率が p のコインを投げるシミュレーションを行う。

S 言語では、r で始まる関数は、特定の確率分布に従う疑似乱数を発生させる。コイン投げは、1 回ごとの試行はベルヌーイ試行と呼ばれる。そして n 回の試行のうちの表が出る回数 k は、各試行で表が出る確率を p として、二項分布 $B(n, p)$ に従う、と言う。その確率関数は

$$p(k) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k} \tag{1}$$

であり、表が出る回数の期待値や分散は、簡単な計算問題であることが、講義や教科書などで紹介されているはずである。

以下では、n 回の試行のうちの成功回数 (表が出た回数) を k、そしてそれらの試行の繰り返し回数 (ここではセットと呼ぶ) を m と記す。

3.1 二項分布のシミュレーション

以下の命令を複数回ずつ実行して、ベルヌーイ試行と二項分布について、復習せよ。

```
> rbinom(n=1,size=1,prob=1/2)
```

を 1 回実行すると、コインを 1 回だけ投げる確率シミュレーションを 1 セット、行うことができる。

```
> rbinom(n=1,size=3,prob=1/2)
```

とすると、$n = 3$ 回投げて表が出た回数を得る確率シミュレーションを $m = 1$ セット、行う。

```
> rbinom(n=10,size=3,prob=1/2)
```

とすると、$n = 3$ 回投げて表が出た回数を得る確率シミュレーションを $m = 10$ セット、行う。

次に、$m = 100$ セットのシミュレーション実験を行い、その結果を図示してみる。

```
> x <- rbinom(n=100,size=1,prob=1/2)
> plot(x,type="l")
```

[2]疑似乱数の 0 から $2^n - 1$ までの値の頻度が、ただ一様なだけでは、様々な目的に用いるのには不十分である。そのため、様々な一様性が定義され、またそれらを保証する疑似乱数が発明されてきた。それらの詳細を紹介することは、この講義の範囲を大幅に逸脱するため、省略するが、興味のある人は参考文献に挙げた三冊の書籍 [1-3] を参考にされたい。

グラフから、作為的に表のみ ($x_i = 1$) もしくは、裏のみ ($x_i = 0$) が連続して発生する、あるいは表と裏が交互に現れる、などのパターンが読み取れるだろうか。

さて、平均と分散も計算し、シミュレーション結果のヒストグラムに図示してみる。

```
> hist(x)
> lines(c(mean(x),mean(x),mean(x), mean(x)+sqrt(var(x))),
+       c(0,max(hist(x)$counts),max(hist(x)$counts)/2,
+         max(hist(x)$counts)/2))
```

二行目はヒストグラムの上に、(平均,0)、(平均, ヒストグラムの高さ)、(平均、ヒストグラムの高さの半分)、(平均+標準偏差, ヒストグラムの高さの半分) という 4 点を順に結んだ折れ線を描いている。よって、グラフに追加して描いた縦棒は、全観測値の平均の値の示し、横棒は分散の平方根 (標準偏差) を表す。

課題 1: 二項分布 $B(n, p)$ について、試行回数 n と成功確率 p 、およびシミュレーションのセット回数 m を与えたら、乱数のグラフと、ヒストグラムを、ひとつの後述するグラフィック・デバイスに描画する関数を作成せよ。それを用いて、試行回数は $n = 2, 5, 10, 100$ の各場合, 成功確率が $p = 0.2, 0.5, 0.8$ の各場合、の組み合わせで、疑似乱数を、$m = 1000$ セット発生させるシミュレーションを行い、その乱数のグラフ (`plot()`) と、ヒストグラム (`hist()`) を描画せよ。そして、n の変化や p の変化に応じて、ヒストグラムがどう変化するかを、論じること。

3.2 グラフの保存について

Windows 版の R では、グラフ描画の関数を実行すると、自動的にサブウィンドウが開かれる。これらをグラフィックス・デバイスと言う。明示するには、次の関数を用いる。

```
windows(height=縦方向の長さ, width=横方向の長さ)
   # Windows 版の R で、画面表示用の Graphics Device を開く
   # どちらもデフォルトは 7
   # 長さの単位は inch(インチ, 1 inch = 2.54 cm)
```

このグラフィックス・デバイスに描いたグラフは、

1. 右クリックで表示されるコンテキストメニューから〔メタファイルにコピー〕を実行すると、そのまま Word などに貼り付けることができる
2. 〔メタファイルに保存〕を実行して、保存すると、ファイルを持ち帰って、自宅でレポート作成作業を続けることができる
3. 〔ポストスクリプトファイルに保存〕を実行して、保存すると、LaTeX でレポートを作成している人には役に立つ

などの用途に用いられる。またグラフィックス・デバイスは、1 枚だけでなく、同時に複数枚のグラフを表示することができる。

```
par(mfrow=c(縦の分割数, 横の分割数))
    # グラフの描画領域を縦横に分割し、多くのグラフを表示させる
```

また、グラフを何枚も表示させると小さくなりますが、字は変わらない。字がグラフの小ささに見合わないときには、

```
par(cex=倍率)
    # 文字の大きさを、倍率をかけて変更する
```

というオプションも設定することがある。

これらのグラフィック・デバイスを開く関数と、そのオプションの用法は例えば次の通り。

```
> windows(height=8, width=8)
> par(mfrow=c(3,3), cex=0.5)
> for( i in c(2:10) ) {
+   hist(rbinom(n=1000, size=i, prob=1/2))
+ }
```

成功確率が $p = 0.5$, 試行回数は $n = 2 \sim 10$ の場合のシミュレーションを、$m = 1000$ セット実行した結果のヒストグラムを、グラフィック・デバイスを 9(= 3 × 3) 分割して描く例である。

課題 1 に関連する図表は、

```
> windows(height=4, width=8)
> par(mfrow=c(1,2), cex=0.7)
> plot(x, sub="二項分布 B(10,0.5)", xlab="シミュレーション回数", ylab="二項乱数")
> hist(x)
```

あるいは

```
> windows(height=8, width=4)
> par(mfrow=c(2,1), cex=0.7)
> plot(x)
> hist(x)
> lines(c(mean(x),mean(x),mean(x), mean(x)+sqrt(var(x))),
+       c(0,max(hist(x)$counts),max(hist(x)$counts)/2,
+         max(hist(x)$counts)/2))
```

かもしれない。レポートに貼り付ける様式、で描くと良い。

3.3 二項分布の平均と分散

先のシミュレーションは、特定の n と p の組み合わせについての結果である。今度は、n を固定し、p を 0 に近い値から 1 に近い値まで変化させたときの、二項分布の平均と分散の関係を調べる。そのために、二項分布に従う疑似乱数を用いて、その平均や分散を推定する。

まずセット数 m を定義し、p を変化させるときの、p の値を、予めリストに代入しておく。

```
> m <- 1000
> n <- 2
> p.binom <- c(1:99)/100
```

次に、p の各値ごとに、$m = 1000$ セットのシミュレーションを実施し、それらの平均や分散を計算するために、平均と分散を納めるリストを作成する。

```
> mean.binom <- vector(mode="numeric",
+                      length=length(p.binom) )
> var.binom  <- vector(mode="numeric",
+                      length=length(p.binom) )
```

あとは、シミュレーションを実施するだけ。

```
> x.temp <- rbinom(n=m, size=n, prob=p.binom[1]
> mean.binom[1] <- mean(x.temp)
> var.binom[1]  <- var(x.temp)
```

以上を、すべての p について実行すると、

```
> plot(mean.binom, var.binom, type="l")
```

で、平均と分散の散布図が描ける。

課題 2: 二項分布 $B(n, p)$ について、試行回数 n と成功確率 p 、およびシミュレーションのセット回数 m を与えたら、m セット分の平均値と分散の折れ線グラフを描く関数を作成せよ。乱数のグラフと、ヒストグラムを、ひとつの後述するグラフィック・デバイスに描画する関数を作成せよ。それを用いて、試行回数は $n = 2, 5, 10, 100$ の各場合, 成功確率が $p = 0.01 \sim 0.99$ まで変化させ、シミュレーション回数は $m = 100, 1000$ セットの二通りを行うこと。そして、n の変化や m の変化を考慮しながら、平均と分散の関係を論じよ。