Annales Academiae Paedagogicae Cracoviensis

Folia 5
Studia Ad Calculum Probabilitatis
Eiusque Didacticam Pertinentia I (2002)

Jerzy Ombach

Komputerowy program Maple[1] w nauczaniu rachunku prawdopodobieństwa

Abstract. The aim of this paper is to establish a few examples of how Maple, one of the up-to-date computer algebra systems, can be used in a process of teaching of the calculus of probability. After some general remarks and comments on Maple we discuss: rolling the dice (false die including), drawings, waiting times, geometric probability, Markov chains, Central Limit Theorem and Law of Large Numbers.

W ostatnich latach opracowano i wprowadzono do powszechnej sprzedaży kilka programów komputerowych określanych wspólną etykietą: *Computer Algebra System* (CAS). Otworzyły się tym samym zupełnie nowe możliwości dla pracowników nauki: matematyków, przyrodników, inżynierów, a także specjalistów z innych dziedzin. Z drugiej strony, posługiwanie się tymi programami zostało już włączone do kanonu wielu kierunków uniwersyteckich na całym świecie, a także na niektórych kierunkach w Polsce. Pojawiły się pierwsze wersje programów typu CAS dla uczniów. Należy się liczyć z jeszcze szybszym rozwojem i upowszechnianiem CAS w najbliższych latach i zdać sobie sprawę, że dostajemy do ręki zupełnie nowy środek dydaktyczny, który może zrewolucjonizować nauczanie matematyki i który stanowi tym samym poważne wyzwanie na początku XXI wieku.

Celem tej pracy jest krótkie przedstawienie charakterystycznych możliwości CAS ze szczególnym uwzględnieniem zastosowania CAS w nauczaniu rachunku prawdopodobieństwa na poziomie szkoły średniej i szkoły wyższej. Autor ma nadzieję, że praca ta pomoże Czytelnikowi zrobić pierwszy krok w kierunku samodzielnej twórczej pracy z pomocą CAS.

Istnieje kilka programu typu CAS. Dla ustalenia uwagi, będziemy zajmować się jednym z nich — programem Maple. Subiektywnym zdaniem autora jest to obecnie najlepszy program w tej klasie.

---

[1] Maple jest zastrzeżonym znakiem towarowym University of Waterloo.

Maple jest produktem Waterloo Maple, Inc. z Kanady i obok programu Mathematica[2] firmy Wolfram Research, Inc. z USA jest przypuszczalnie najczęściej na świecie używanym programem do obliczeń symbolicznych. W Polsce, bardziej popularna jest jak na razie Mathematica, jakkolwiek moim zdaniem, tendencja ta zaczyna ulegać zmianie. Inna sprawa, że kolejne wersje obydwu programów przejmują od konkurenta sprawdzone rozwiązania i tym samym stają się coraz bardziej do siebie podobne.

Poniżej przytoczone przykłady zostały wykonane przy pomocy programu Maple V wersja 5, która weszła do sprzedaży w pierwszym kwartale 1998 roku. Warto tutaj nadmienić, że jest to program wyjątkowo przyjazny dla początkującego użytkownika (znającego choćby w niewielkim stopniu język angielski). Już po kilku minutach pracy można rozwiązywać układy równań, obliczać pochodne i całki, rysować wykresy .... Aby jednak w pełni wykorzystać inne potencjalne możliwości Maple, powinno się spędzić trochę czasu na pracę bezpośrednio przy komputerze i wtedy też warto przejrzeć niektóre spośród ponad 200 pozycji poświęconych bezpośrednio lub pośrednio temu programowi. Odpowiednia lista jest dostępna na stronie internetowej Maple.

W następnych punktach przytoczymy kolejno: kilkanaście prostych przykładów ilustrujących zastosowanie Maple, wskażemy kilka charakterystycznych właściwości tego programu, omówimy kilka typowych sytuacji probabilistycznych; pokazując sposób zastosowania Maple do ich analizy i w końcu, omówimy bardziej złożone zadanie probabilistyczne.

1. Przykłady wprowadzajce

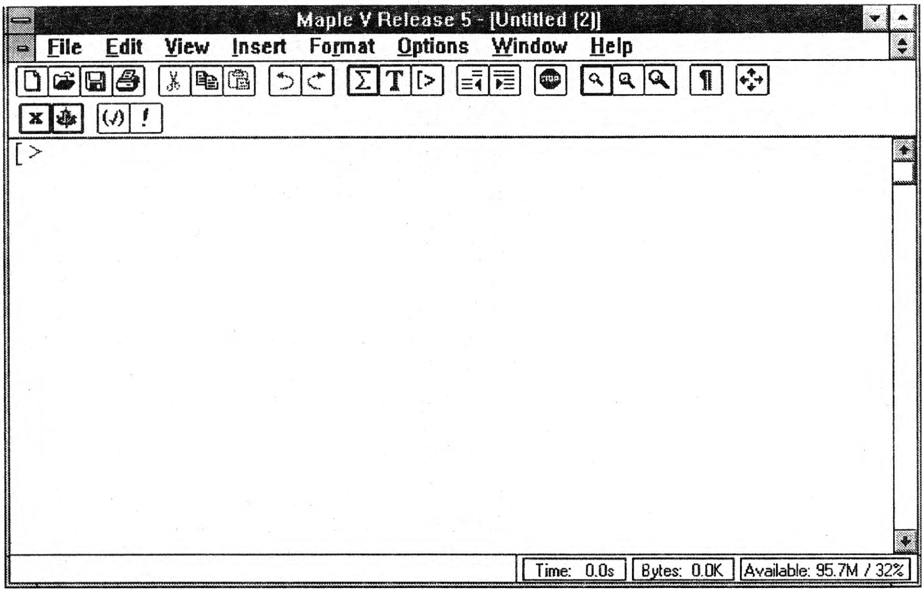
Pisząc tę pracę, wykorzystano Maple V release 5[3] działającą w środowisku Windows 95. Wersja ta działa także we wszystkich innych popularnych systemach operacyjnych. Instalacja jest bardzo prosta i nie wymaga żadnych „sztuczek".

Uruchommy w standardowy sposób Maple V Release 5. Jeżeli jest to na naszym komputerze pierwszy start tej wersji Maple, dostajemy propozycję odbycia wycieczki po systemie celem wstępnego zaznajomienia się z jego możliwościami. Warto z tej propozycji skorzystać. Przy następnych startach nasz ekran może wyglądać tak jak na rysunku 1.

Maple otwarła dla nas tak zwany `worksheet`, czyli stronę do pracy, na której możemy pisać komendy i oglądać wyniki. Mamy przy tym do dyspozycji cały zestaw narzędzi do pracy z plikami, drukowania, edycji, formatowania, ustawiania opcji i korzystania z pomocy. Na dole, w prawym rogu wyświetlane są aktualne informacje o systemie. W lewym górnym rogu białego obszaru

[2] Mathematica jest znakiem towarowym Wolfram Research, Inc.
[3] Maple 7 z lipca 2001 r. działa podobnie

strony znajduje się znak zachęty [>. Tam też znajduje się kursor. Możemy rozpocząć pracę



Rysunek 1.

Pisząc w linii komend — zwróćmy uwagę na średnik;

```
> (5/4 - 2/7)/(1 - 5/2)^2;
```

otrzymamy:

$$\frac{3}{7}.$$

Podstawową własnością CAS jest praca z wyrażeniami algebraicznymi. Wprowadźmy takie wyrażenie:

```
> 1/(a-b)^2+2/(a+b)^2 - 1/(a^4 - b^4);
```

$$\frac{1}{(a-b)^2}+2\,\frac{1}{(a+b)^2}-\frac{1}{a^4-b^4}$$

Może można to wyrażenie uprościć (znak procentu % oznacza odwołanie się do ostatniego wyniku)?

```
> simplify(%);
```

$$\frac{3\,a^4-2\,a^3\,b-a^2+6\,a^2\,b^2-2\,a\,b^3+3\,b^4+b^2}{(a^4-b^4)\,(a+b)\,(a-b)}$$

Podstawmy za a oraz b wartości liczbowe:

```
> subs((a=2,b=1),%);
```

$$\frac{52}{45}$$

Zamieńmy na ułamek dziesiętny:

```
> evalf(%);
```

$$1.155555556$$

Standardowo Maple używa 10 cyfr znaczących. Można to zmienić:

```
> evalf(Pi,100);
```

$$3.141592653589793238462643383279502884197169399375105 8\backslash$$
$$2097494459230781640628620899862803482534211706 8$$

Przypomnijmy sobie wartości pewnych funkcji:

```
> cos(Pi/3);
```

$$\frac{1}{2}$$

```
> arcsin(1/2);
```

$$\frac{1}{6}\pi$$

...

Rozwiążmy równanie algebraiczne stopnia trzeciego:

```
> solve(2*x^3 - 3*x^2 + 12*x - 5,x);
```

$$-\frac{1}{2}\%2+\frac{7}{2}\%1+\frac{1}{2},\ \frac{1}{4}\%2-\frac{7}{4}\%1+\frac{1}{2}+\frac{1}{2}I\sqrt{3}\left(-\frac{1}{2}\%2-\frac{7}{2}\%1\right),$$
$$\frac{1}{4}\%2-\frac{7}{4}\%1+\frac{1}{2}-\frac{1}{2}I\sqrt{3}\left(-\frac{1}{2}\%2-\frac{7}{2}\%1\right)$$
$$\%1 := \frac{1}{(1+2\sqrt{86})^{(1/3)}}$$
$$\%2 := (1+2\sqrt{86})^{(1/3)}$$

Otrzymaliśmy jedno rzeczywiste i dwa istotnie zespolone rozwiązania. Poznajmy ich przybliżone wartości:

```
> evalf(%,3);
```

$$.450,\ .522-2.29\,I,\ .522+2.29\,I$$

Spróbujmy rozwiązać inne równanie:

```
> solve(2*sin(x^2) = x,x);
```

$$2\sin(\mathrm{RootOf}(\_Z - 4\sin(\_Z)^2))$$

Maple nie umie wyliczyć symbolicznie rozwiązania. A Czytelnik potrafi? Może jednak Maple potrafi to zrobić numerycznie?

```
> evalf(%);
```

$$0$$

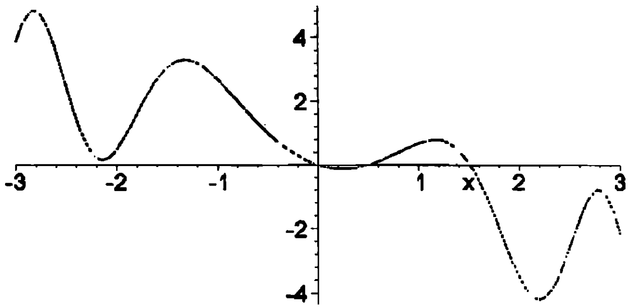
Maple podał jedno z rozwiązań, ale nie o to rozwiązanie przecież nam chodziło. Zastosujmy polecenie `fsolve` numerycznego znajdowania rozwiązania, dając jednocześnie wskazówkę Maple, żeby szukał niezerowego rozwiązania.

```
> fsolve(2*sin(x^2) = x,x,avoid={x=0});
```

$$.5054822723$$

Aby się upewnić, czy są jeszcze inne rozwiązania, narysujmy wykres:

```
> plot(2*sin(x^2)-x,x=-3..3);
```



Rysunek 2.

Więc jednak jest jeszcze jedno. Dajmy Maple wskazówkę, że ma go szukać w przedziale $[1, 2]$.

```
> fsolve(2*sin(x^2) = x,x=1..2);
```

$$1.511542719$$

. . .

Spróbujmy rozwiązać układ trzech równań. Gdy nie ma wątpliwości, co jest zmienną, nie musimy nawet zmiennych specyfikować.

```
> solve({x^2 + 2*y^2 + 3*z^2 = 6, z = x^2 + y^2,y=x^3});
```

$$\{y = \mathrm{RootOf}(-\%1 + \_Z^2)\,\%1,\ z = \%1\,(1 + \%1^2),\ x = \mathrm{RootOf}(-\%1 + \_Z^2)\}$$
$$\%1 := \mathrm{RootOf}(\_Z + 2\,\_Z^3 + 3\,\_Z^2 + 6\,\_Z^4 + 3\,\_Z^6 - 6)$$

Ten wynik nie jest zachęcający. Znowu spróbujmy `fsolve`.

```
> fsolve({x^2 + 2*y^2 + 3*z^2 = 6, z = x^2 + y^2,y=2*x^3});
```

$$\{z = 1.180569618,\ x = .7364598458,\ y = .7988720257\}$$

Podobnie jak poprzednio, wykonamy odpowiedni wykres w celu zorientowania się w sytuacji.

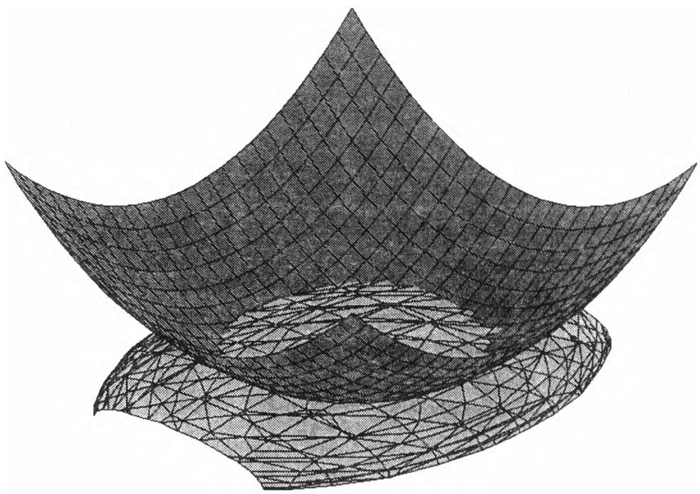
```
> rys1:=plot3d(x^2+y^2,x=-2..2,y=-2..2,color=GREEN):
```

Pisząc na końcu polecenia dwukropek zamiast średnika, poprosiliśmy Maple o niedrukowanie wyniku. Jest on przechowywany pod nazwą `rys1`. Podobnie wykonamy `rys2`. Ponieważ nie chcemy wyliczać z jako funkcji x, y zastosujemy inną komendę:

```
> rys2:=plots[implicitplot3d](x^2 + 2*y^2 + 3*z^2 = 6,
  x=-2..2,y=-2..2,z=0..2,color=BLUE):
```

Zobaczmy dotychczasowy wynik:

```
> plots[display](rys1,rys2);
```



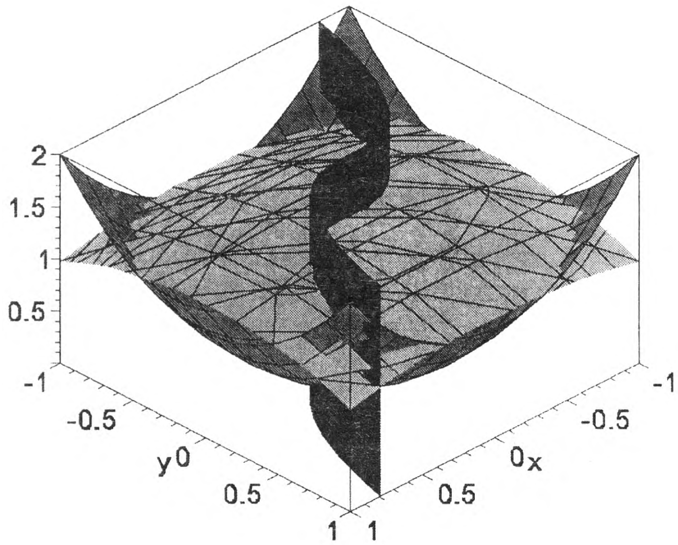
Rysunek 3.

Do narysowania ostatniej powierzchni wykorzystamy jej równania parametryczne:

```
> rys3:=plot3d([t,2*t^3,s],t=-1..1,s=0..2,color=RED):
```

Jesteśmy gotowi do wykonania ostatecznego wykresu:

```
> plots[display](rys1,rys2,rys3,view=[-1..1,-1..1,0..2],
  axes=BOXED);
```



Rysunek 4.

Jest więc jeszcze jedno rozwiązanie i dość łatwo je zlokalizować:

```
> fsolve({x^2 + 2*y^2 + 3*z^2 = 6, z = x^2 + y^2,y=2*x^3},
  {x=-1,y=-1,z=2});
```

$$\{z = 1.180569618, x = -.7364598458, y = -.7988720257\}$$

2. Cechy charakterystyczne programu Maple

Powyższe przykłady ukazują jedynie niewielki procent potencjalnych możliwości Maple. Nie jest możliwe wymienienie ich wszystkich tutaj. Nie robią nawet tego podręczniki dostarczane przez producenta [2], [3]. W bibliografii polskiej, Maple dotyczą częściowo pozycje [6], [5] oraz [4]. Zainteresowani powinni zaznajomić się z pewnością z rozbudowanym i wygodnym w użyciu systemem pomocy Help. Zaczniemy właśnie od omówienia tego systemu.

System pomocy – Help

Jednym z mocnych punktów Maple jest system pomocy. Powiedzmy od razu, że działa on trochę inaczej — lepiej moim zdaniem — niż analogiczne systemy w typowych programach pracujących na platformie Windows. Głównym składnikiem są strony omawiające wszystkie polecenia Maple. Do interesującej nas strony można się dostać na kilka różnych sposobów: spisy tematyczne, alfabetyczne, indeksy lub pomoc kontekstowa. Oprócz dość formalnej składni i rozsądnego omówienia danego polecenia, na stronie zamieszczone są dość proste przykłady jego użycia. Przykłady te można łatwo skopiować, modyfikować i natychmiast uruchamiać. Dzięki hiperłączom oraz innym udogodnieniom, można też bezpośrednio przechodzić do innych stron tematycznie związanych. W danym momencie może być otwartych wiele stron pomocy.

Pakiety

Po uruchomieniu, program Maple oferuje nam pewną liczbę najczęściej używanych funkcji i poleceń. Większość jednak zasobów Maple nie jest jeszcze bezpośrednio dostępna. Są one podzielone na pakiety i aby skorzystać z danej komendy trzeba ją wywołać używając nazwę pakietu, w którym się znajduje. W ten właśnie sposób w poprzednim punkcie skorzystaliśmy z poleceń `implicitplot3d` oraz `display` znajdujących się w pakiecie `plots`. Jeżeli jednak chcemy użyć więcej komend z tego samego pakietu, można go wczytać w całości używając polecenia `with`.

Dla wyrobienia pewnej orientacji przytaczamy przepisaną z systemu Help listę wszystkich pakietów[4] (pakiet — `package`) dostępnych w wersji 5. Rozszerzają one istotnie możliwości systemu[5].

The following packages are available:

DEtools	differential equations tools
Domains	create domains of computation
GF	Galois Fields
GaussInt	Gaussian Integers
LREtools	manipulate linear recurrence relations
Matlab	Matlab Link
algcurves	Algebraic Curves
codegen	Code Generation

[4] Na liście tej jednak brak pakietu `PDEtools` służącego do rozwiązywania i przekształcania równań różniczkowych cząstkowych. Pakiet faktycznie jest dostępny.

[5] Zdecydowana większość pakietów jest dostępna na wszystkich popularnych platformach.

combinat	combinatorial functions
combstruct	combinatorial structures
diffalg	differential algebra
difforms	differential forms
finance	financial mathematics
genfunc	rational generating functions
geom3d	Euclidean three-dimensional geometry
geometry	Euclidean geometry
grobner	Grobner bases
group	permutation and finitely-presented groups
inttrans	integral transforms
liesymm	Lie symmetries
linalg	Linear algebra
logic	Boolean logic
networks	graph networks
numapprox	numerical approximation
numtheory	number theory
orthopoly	orthogonal polynomials
padic	p-adic numbers
plots	graphics package
plottools	basic graphical objects
powseries	formal power series
process	(Unix)-multi-processing
simplex	linear optimization
stats	statistics
student	student calculus

sumtools	indefinite and definite sums
tensor	tensor computations and General Relativity
totorder	total orders on names

- For information see ?package where package is from the above list. This will give a list of the functions available in the package. To cause all functions in a package to be defined in the session, do: with(package);
- For information on a particular package function, see ?package,function

Podkreślenia wyrazów oznaczają możliwość uzyskania dodatkowych informacji o danym haśle — wystarczy kliknąć to hasło myszką, aby pojawiła się odpowiednia strona pomocy.

Aby korzystać z wybranego pakietu wystarczy go wczytać poleceniem `with`. Na przykład:

```
> with(linalg);

Warning, new definition for norm

Warning, new definition for trace
```

[*BlockDiagonal*, *GramSchmidt*, *JordanBlock*, *LUdecomp*, *QRdecomp*, *Wronskian*, *addcol*, *addrow*, *adj*, *adjoint*, *angle*, *augment*, *backsub*, *band*, *basis*, *bezout*, *blockmatrix*, *charmat*, *charpoly*, *cholesky*, *col*, *coldim*, *colspace*, *colspan*, *companion*, *concat*, *cond*, *copyinto*, *crossprod*, *curl*, *definite*, *delcols*, *delrows*, *det*, *diag*, *diverge*, *dotprod*, *eigenvals*, *eigenvalues*, *eigenvectors*, *eigenvects*, *entermatrix*, *equal*, *exponential*, *extend*, *ffgausselim*, *fibonacci*, *forwardsub*, *frobenius*, *gausselim*, *gaussjord*, *geneqns*, *genmatrix*, *grad*, *hadamard*, *hermite*, *hessian*, *hilbert*, *htranspose*, *ihermite*, *indexfunc*, *innerprod*, *intbasis*, *inverse*, *ismith*, *issimilar*, *iszero*, *jacobian*, *jordan*, *kernel*, *laplacian*, *leastsqrs*, *linsolve*, *matadd*, *matrix*, *minor*, *minpoly*, *mulcol*, *mulrow*, *multiply*, *norm*, *normalize*, *nullspace*, *orthog*, *permanent*, *pivot*, *potential*, *randmatrix*, *randvector*, *rank*, *ratform*, *row*, *rowdim*, *rowspace*, *rowspan*, *rref*, *scalarmul*, *singularvals*, *smith*, *stackmatrix*, *submatrix*, *subvector*, *sumbasis*, *swapcol*, *swaprow*, *sylvester*, *toeplitz*, *trace*, *transpose*, *vandermonde*, *vecpotent*, *vectdim*, *vector*, *wronskian*]

Otrzymaliśmy dwa rodzaje informacji. Ostrzeżenia (warming), że od tej chwili obowiązuje zmieniona definicja normy (`norm`) i śladu (`trace`) macierzy. Po drugie, dostaliśmy kompletną listę poleceń zawartych w naszym pakiecie. Aby teraz poznać działanie i sposób użycia interesującego nas polecenia, można napisać ?NazwaPolecenia.

Możemy teraz używać wszystkich powyższych procedur, jak i wszystkich procedur standardowych, ewentualnie procedur z wczytanych wcześniej innych pakietów:

Zadajemy dla przykładu macierz o pięciu wierszach i czterech kolumnach:

```
> A := array( [[1,2,3,4],[4,3,2,1],[1,1,1,1],
  [2,2,2,2],[-4,5,7,1]] );
```

$$A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ -4 & 5 & 7 & 1 \end{bmatrix}$$

Transponujemy:

```
> A:=transpose(A);
```

$$A := \begin{bmatrix} 1 & 4 & 1 & 2 & -4 \\ 2 & 3 & 1 & 2 & 5 \\ 3 & 2 & 1 & 2 & 7 \\ 4 & 1 & 1 & 2 & 1 \end{bmatrix}$$

Obliczamy rząd A:

```
> rank(A);
```

$$3$$

Wyznaczamy bazę jądra odwzorowania liniowego $x \longrightarrow Ax$:

```
> kernel(A);
```

$$\{[1, 1, -5, 0, 0], [0, 0, -2, 1, 0]\}$$

Sprawdzamy, stosując pożyteczną komendę `map`, czy rzeczywiście wskazane wektory należą do jądra:

```
> map(x->multiply(A,x),%);
```

$$\{[0, 0, 0, 0]\}$$

. . .

Dla Czytelnika tej pracy najważniejszy jest pakiet `stat`. Pisząc:

```
> with(stats);
```

otrzymujemy:

[*anova*, *describe*, *fit*, *importdata*, *random*, *statevalf*, *statplots*, *transform*]

Ten pakiet składa się z kilku mniejszych pakietów właśnie wymienionych. Dla przykładu:

```
> with(random);
```

[β, *binomiald*, *cauchy*, *chisquare*, *discreteuniform*, *empirical*, *exponential*, *fratio*, γ, *laplaced*, *logistic*, *lognormal*, *negativebinomial*, *normald*, *poisson*, *studentst*, *uniform*, *weibull*]

Share

Oprócz pakietów producent dostarcza kilkaset programów w tak zwanym systemie Share. Są to programy pisane przez użytkowników Maple i w pewnym sensie autoryzowane przez Maple. Dotyczą one wielu szczegółowych problemów matematycznych, technicznych, a niektóre są po prostu grami komputerowymi. Informacje o tych programach można znaleźć w systemie Help. Dla przykładu przytaczamy informację o jednym z programów z biblioteki Share:

mark

- Author: Lang, Jerome <jmlang@maplesoft.com>.
- worksheet showing many examples of the new statistics package (note: worksheet needs to be able to read the file 'mark.dat'
- Help Links: share[mark] share[mark,mark_mws] with share share[index,author] share[index,subject] share[index,alpha]
- Related Packages: spectral, mark, random.

Other packages by the same author: random

Dalsze możliwości Maple

Oprócz wspomnianych powyżej, Maple posiada szereg innych możliwości. Na przykład, są to: możliwość animacji, podręczne palety z symbolami matematycznymi do graficznego tworzenia komend, tłumaczenie całych dokumentów na język $\LaTeX2\varepsilon$ i na format HTML, eksportowanie grafiki w wielu formatach i wbudowane arkusze kalkulacyjne (jeszcze dość prymitywne). Wyjątkowo ważnym atutem Maple jest bardzo wygodny w użyciu system pomocy. Można w związku z tym całkiem swobodnie korzystać z Maple nie zaglądając do żadnej książki. Być może dlatego, podręczniki dostarczane przez producenta, jakkolwiek poprawnie napisane i miłe w czytaniu, nie zawierają wielu istotnych informacji.

Jak każdy złożony system komputerowy, Maple nie jest pozbawiony pewnych usterek i z pewnością niektóre rozwiązania powinny być udoskonalone lub zmienione w następnych wersjach. Warto jednak wspomnieć, że Maple pracuje całkiem dobrze w środowisku Windows 95 i, w przeciwieństwie do niektórych podobnych programów, w razie wystąpienia problemów nie zawiesza całego systemu.

3. Maple i elementarny rachunek prawdopodobieństwa

Podamy teraz przykłady typowych problemów dotyczących elementarnego rachunku prawdopodobieństwa, które mogą być omawiane z użyciem Maple. Będziemy przy tym podkreślać bardziej te aspekty, które dotyczą techniki stosowania Maple.

. . .

Przeprowadźmy symulację 100 krotnego rzutu dwiema kostkami symetrycznymi, przy czym interesuje nas jedynie suma oczek uzyskanych na obu kostkach.

```
> kostka1 := rand(1..6):
  kostka2 := rand(1..6):
  wyniki := NULL:
  from 1 to 100 do
  wyniki := wyniki,kostka1() + kostka2()
  od:
  wyniki;
```

5, 4, 8, 6, 3, 8, 7, 2, 6, 3, 9, 9, 7, 7, 9, 5, 10, 5, 6, 9, 6, 12, 3, 6, 6, 6, 7, 5, 6, 5, 9, 7, 7, 7, 12, 11, 4, 4, 6, 7, 11, 9, 10, 10, 5, 3, 6, 9, 7, 8, 8, 6, 6, 4, 6, 12, 11, 9, 5, 8, 10, 5, 5, 5, 7, 8, 10, 7, 6, 8, 6, 9, 6, 7, 10, 6, 5, 9, 12, 5, 6, 4, 8, 8, 10, 9, 6, 7, 7, 8, 7, 6, 7, 4, 8, 5, 4, 12, 5, 8

Zobaczmy, jak często wypadają „środkowe" liczby 6, 7, 8 w porównaniu z liczbami „skrajnymi" 2 i 12.

```
> A := {6,7,8}: B:= {2,12}:

> nA := 0: nB := 0:
  for i from 1 to 3600 do
  suma := kostka1() + kostka2():
  if member(suma,A) then nA := nA + 1 fi:
```

```
    if member(suma,B) then nB := nB + 1 fi
    od:
> nA,nB,nB/nA;
```

$$1613,\ 192,\ \frac{192}{1613}$$

Ostatnia liczba, jak łatwo sprawdzić, powinna teoretycznie wynosić $\frac{1}{8}$. Czy różnica jest duża?

```
> evalf(%[3] - 1/8);
```

$$-.005967141971$$

. . .

Można „rzucać" kostką niesymetryczną. Zwykle na takich kostkach „1" pojawia się najczęściej!

```
> kostka := empirical[0.26,0.15,0.15,0.15,0.15,0.14]:
> stats[random,kostka](10);
```

$$1.0,\ 5.0,\ 4.0,\ 1.0,\ 6.0,\ 2.0,\ 1.0,\ 4.0,\ 2.0,\ 6.0$$

Pozbędziemy się miejsc dziesiętnych:

```
> wyniki :=map(trunc,[%]);
```

$$wyniki := [1,\ 5,\ 4,\ 1,\ 6,\ 2,\ 1,\ 4,\ 2,\ 6]$$

Wykonajmy 1000 rzutów niesymetryczną kostką:

```
> stats[random,kostka](1000):
> wyniki :=map(trunc,[%]):
```

Ile razy wypadały poszczególne ścianki?

```
> for i from 1 to 6 do
  w:=x->(x=i):
> a.i:=nops(select(w,wyniki,x))
  od:
> seq(a.i,i=1..6);
```

$$236,\ 139,\ 138,\ 150,\ 168,\ 169$$

. . .

Wiemy jak obliczyć prawdopodobieństwo wygrania czwórki w grze liczbowej, gdzie skreślamy 7 spośród 49 liczb, natomiast maszyna losująca losuje 6 liczb.

```
> binomial(7,4)*binomial(42,2)/binomial(49,6);
```

$$\frac{205}{95128}$$

```
> evalf(%);
```

$$.002154991170$$

Przeprowadźmy symulację tej gry. Najpierw skreślamy liczby, a później Maple zastępuje maszynę:

```
> nasze_liczby :={7,23,24,28,29,36,41}:
> with(combinat);
Warning, new definition for Chi
```

[Chi, *bell*, binomial, *cartprod*, *character*, *choose*, *composition*, *conjpart*, *decodepart*, *encodepart*, *fibonacci*, *firstpart*, *graycode*, *inttovec*, *lastpart*, *multinomial*, *nextpart*, *numbcomb*, *numbcomp*, *numbpart*, *numbperm*, *partition*, *permute*, *powerset*, *prevpart*, *randcomb*, *randpart*, *randperm*, *stirling1*, *stirling2*, *subsets*, *vectoint*]

```
> wygrane := randcomb(49,6);
```

$$wygrane := \{19, 21, 25, 31, 40, 48\}$$

Przeprowadźmy teraz 20 serii po 100 losowań i zobaczmy, ile razy wygramy dwójkę.

```
> for j from 1 to 20 do
> dwójki := 0:
> from 1 to 100 do
> wygrane := randcomb(49,6):
> if nops(nasze_liczby intersect wygrane) = 2 then
  dwojki := dwojki + 1
  fi
  od:
  lista.j := dwojki:
> od: seq(lista.j,j=1..20);
```

14, 14, 18, 11, 15, 16, 14, 11, 11, 15, 19, 16, 24, 17, 16, 16, 19, 18, 12, 14

A jak teoretycznie powinno być?

```
> binomial(7,2)*binomial(42,4)/binomial(49,6);
```

$$\frac{7995}{47564}$$

```
> evalf(%);
```

$$.1680893112$$

Jaka jest zgodność wyniku eksperymentu z teorią?

```
> evalf(1/20*add(lista.j, j = 1..20));
```

$$15.50000000$$

Zgodność nie jest w tym przypadku najlepsza — sprawa jej oceny należy już jednak do statystyki.

. . .

Jakie jest prawdopodobieństwo, że w 20 osobowej klasie znajdują się co najmniej dwie osoby obchodzące w tym samym dniu swoje urodziny. Jest to klasyczny problem, w którym łatwo jest wypisać wzór, lecz trudniej z niego skorzystać w celu otrzymania efektywnego wyniku.

```
> binomial(365,20)*20!/365^20;
```

$$\frac{45443978228710616760289446219108390846914887 68}{77211929831874031340970916361211101379394531 25}$$

```
> evalf(%);
```

$$.5885616164$$

```
> 1 -%;
```

$$.4114383836$$

Powyższy wynik może się wydawać na tyle ciekawy, że powtórzymy nasze obliczenia dla klas o innej liczbie uczniów. Wykorzystujemy w tym celu wbudowany arkusz kalkulacyjny.

k	$\frac{\text{binomial}(365,\, k) * k!}{365^k}$	$1 - \frac{\text{binomial}(365,\, k) * k!}{365^k}$
5	.9728644263	.0271355737
10	.8830518223	.1169481777
20	.5885616164	.4114383836
50	.02962642042	.9703735796
100	$.3072489279\, 10^{-6}$	.9999996928

Tabela 1.

Dla sprawdzenia, wylosujmy daty (jako kolejne din roku) urodzin uczniów w hipotetycznej 20 osobowej klasie.

```
> klasa := NULL:
> data :=rand(1..365):
  for i from 1 to 20 do
  klasa := klasa,data()
  od:
  klasa;
```

126, 120, 127, 301, 161, 144, 20, 129, 50, 274, 277, 39, 336, 71, 304, 265, 95, 138, 137, 218

Jeżeli nie dowierzamy naszej spostrzegawczości, poprośmy Maple, aby sprawdził, ile jest różnych dat.

```
> nops({klasa});
```

20

Przyglądnijmy się innym 10 klasom:

```
> proba := NULL:
  from 1 to 10 do
  klasa := NULL:
  from 1 to 20 do
  klasa := klasa,data()
  od:
  proba := proba,nops({klasa})
  od:
  proba;
```

20, 20, 20, 20, 20, 19, 20, 20, 19, 19

...

Jak długo czeka się na pojawienie się „6" przy grze w Chińczyka?

```
> kostka := rand(1..6):
> from 1 to 10 do
  rzuty := NULL:
> oczka := 0:
> while oczka <> 6 do
  oczka := kostka():
  rzuty := rzuty,oczka
  od:
  print(rzuty);
  od:
```

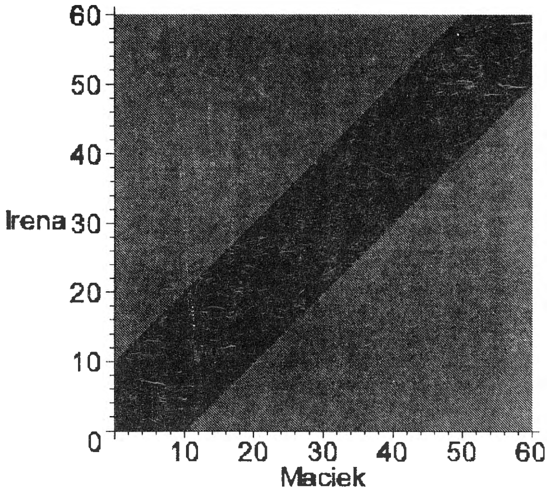
1, 2, 5, 1, 6
1, 6
6
6
1, 5, 1, 4, 3, 1, 6
1, 2, 3, 3, 1, 3, 6
4, 5, 3, 1, 1, 5, 5, 3, 1, 3, 5, 2, 3, 1, 3, 3, 1, 3, 6
2, 5, 6
4, 5, 1, 2, 5, 2, 1, 6
1, 5, 2, 5, 3, 3, 1, 2, 5, 6

...

Irena i Maciek umówili się na rozmowę w internecie między godziną 22 a 23 czasu polskiego, przy czym będą na siebie czekać do 10 minut. Jakie jest prawdopodobieństwo ich spotkania?

Zróbmy klasyczny do tego zadania rysunek.

```
> with(plots):
> inequal( {Irena - Maciek < 10,Maciek - Irena <10},
  Irena=0..60, Maciek=0..60, optionsfeasible=(color=red),
  optionsexcluded=(color=green),labels=[Maciek,Irena]);
```



Rysunek 5.

Używając prawdopodobieństwa geometrycznego bez trudu obliczamy, że interesujące nas prawdopodobieństwo wynosi:

```
> 11./36;
```

.3055555556

Zastosowaliśmy tutaj trick. Pisząc kropkę dziesiętną, zmusiliśmy Maple do używania rozwinięć dziesiętnych.

A teraz zróbmy symulację 100 spotkań.

```
> N :=100:
> Irena:=stats[random,uniform[0,60]](N):
> Maciek:=stats[random,uniform[0,60]](N):
> k:= 0:
> for i from 1 to N do
  if abs(Irena[i] - Maciek[i]) < 10 then k:= k+1 fi od:
  k;
```

39

Odświerzmy system:

```
> restart:
```

...

Aby wprowadzić do sprzedaży nowe lekarstwo, każdy producent musi uzyskać zgodę ministerstwa. Producent leku *Friendlymath* uważa, że jeżeli dodatkowe testy nie wykażą istnienia efektów ubocznych, szansa uzyskania zgody wynosi 80%, natomiast w przeciwnym przypadku, szansa na zgodę ministerstwa wynosi jedynie 40%. Producent ocenia też możliwość wykrycia efektów ubocznych na 10%. Jaka jest szansa, że *Friendlymath* pojawi się w aptekach?

Zapiszmy nasze dane:

```
> efekt_uboczny := empirical[0.1,0.9]:
  decyzja_jest_ef := empirical[0.4,0.6]:
  decyzja_brak_ef := empirical[0.8,0.2]:
```

Zdefiniujmy wygodną procedurę:

```
> F:=proc(x)
  if x = 1.0 then jest
  elif x = 2.0 then brak
  fi
  end:
```

Zobaczmy przykład:

```
> N := 15:
> stats[random,efekt_uboczny](N):
  wyniki :=map(F,[%]);
```

$wyniki := [brak, brak, brak, brak, brak, brak, brak, jest, brak, brak, brak, brak, brak, brak, brak]$

```
> for i from 1 to N do
  if wyniki[i] = JEST then
  decyzja[i] := stats[random,decyzja_jest_ef](1) else
  decyzja[i] := stats[random,decyzja_brak_ef](1)
  fi
  od:
> zgoda:=seq(F(decyzja[i]),i=1..N);
```

$zgoda := jest, brak, brak, jest, brak, jest, brak, jest, brak, jest, jest, brak, jest, jest, jest$

```
> w:= x->x=jest:
> nops(select(w,[zgoda],x));
```

$$9$$

```
> %/N;
```

$$\frac{3}{5}$$

Powtórzmy to więcej razy:

```
> N := 1500:
> stats[random,efekt_uboczny](N):
  wyniki :=map(F,[%]):
> for i from 1 to N do
  if wyniki[i] = jest then
  decyzja[i] := stats[random,decyzja_jest_ef](1) else
  decyzja[i] := stats[random,decyzja_brak_ef](1) fi
  od:
> zgoda:=seq(F(decyzja[i]),i=1..N):
> nops(select(w,[zgoda],x));
```

$$1122$$

```
> evalf(%/N);
```

$$.7480000000$$

Korzystając z wzoru na prawdopodobieństwo całkowite, łatwo policzyć, że prawdopodobieństwo uzyskania zgody wynosi $0,76$.

. . .

Wieloletnie doświadczenie wskazuje, że głównym powodem niezdawania egzaminu z rachunku prawdopodobieństwa są niedostateczne wyniki uzyskiwane podczas części pisemnej tego egzaminu. Policzmy prawdopodobieństwo tego, że student, który nie zdał egzaminu, uzyskał ocenę niedostateczną na egzaminie pisemnym, jeżeli: (a) prawdopodobieństwo zdania części pisemnej jest 0,6, (b) aby być dopuszczonym do części ustnej należy uzyskać ocenę pozytywną z części pisemnej, (c) szansę zdania części ustnej ocenia się na 95%.

Zapiszmy nasze dane:

```
> pisemny := empirical[0.6,0.4]:
  ustny_pisemny_z := empirical[0.95,0.05]:
  ustny_pisemny_n := empirical[0.0,1.0]:
```

Zdefiniujmy procedurę o charakterze technicznym:

```
> F:=proc(x)
  if x = 1.0 then zdany elif x = 2.0 then niezdany
  fi
  end:
```

Przeegzaminujmy 1000 studentów:

```
> N := 1000:
> stats[random,pisemny](N):
  wyniki_pisemnego :=map(F,[%]):
> for i from 1 to N do
  if wyniki_pisemnego[i] = zdany then
  ustny[i] := stats[random,ustny_pisemny_z](1) else
  ustny[i] := stats[random,ustny_pisemny_n](1) fi
  od:
> egzamin:=seq(F(ustny[i]),i=1..N):
> w:= x->x=niezdany:
> liczba_niezdanych := nops(select(w,[egzamin],x));
```

$$liczba\_niezdanych := 430$$

```
> %/N;
```

$$\frac{43}{100}$$

Wynik ten dokładnie zgadza się z wynikiem, który można otrzymać stosując wzór na prawdopodobieństwo całkowite, co jest rzadkim przypadkiem. Zobaczmy, ilu studentów nie zdało egzaminu z powodu części pisemnej.

```
> ile:=0:
  for i from 1 to N do
  if egzamin[i] = niezdany
  and wyniki_pisemnego[i] = niezdany
  then ile := ile +1
  fi
  od:
> ile;
```

```
> evalf(ile/liczba_niezdanych);
```

$$.9418604651$$

Stosując natomiast wzór Bayesa'a mamy odpowiednie prawdopodobieństwo warunkowe równe:

```
> 1*0.4/(0.05*0.6+1*0.4);
```

$$.9302325581$$

...

Zdarzenia opisane schematem Bernoulliego można modelować następująco:

```
> F:=proc(x)
  if x = 1.0 then 0 elif x = 2.0 then 1 fi
  end:
> p:=0.2:
> proba := empirical[1-p,p]:
> stats[random,proba](10):
  wyniki :=map(F,[%]);
```

$$wyniki := [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]$$

```
> add(wyniki[i],i=1..nops(wyniki));
```

$$1$$

Policzmy teraz stosunek sukcesów k do liczny prób N, gdy $N = 1000$.

```
> stats[random,proba](1000):
  wyniki :=map(F,[%]):
  evalf(add(wyniki[i],i=1..nops(wyniki))/1000);
```

$$0.1980000000$$

...

Ułóżmy przykładowy model dla jednorodnego łańcucha Markowa o trzech stanach:

```
> P1 := empirical[0.1,0.,0.9]:
  P2 := empirical[0.2,0.1,0.7]:
  P3 := empirical[0.4,0.1,0.5]:
  rozklad_pocz := empirical[0.1,0.8,0.1]:

> T1:= proc()
  trunc(stats[random,P1](1))
  end:
> T2:= proc()
  trunc(stats[random,P2](1))
  end:
```

```
> T3:= proc()
  trunc(stats[random,P3](1))
  end:
```

Jak to działa?

```
> T1(),T2(),T3();
```

$$3, 3, 3$$

Wylosujmy stan początkowy i oglądnijmy jedną z realizacji o długości 150.

```
> stan := trunc(stats[random,rozklad_pocz](1));
```

$$stan := 2$$

```
> sciezka := stan:
  from 1 to 150 do
  stan :=T.stan():
  sciezka := sciezka,stan
  od:
  sciezka;
```

2, 3, 1, 3, 3, 1, 3, 2, 1, 3, 1, 1, 3, 2, 3, 3, 3, 1, 3, 1, 3, 2, 3, 3, 3, 3, 3, 3, 2, 1,
3, 3, 3, 3, 3, 1, 3, 3, 3, 1, 3, 3, 1, 3, 3, 1, 3, 1, 3, 3, 3, 1, 3, 3, 3, 1, 3,
2, 3, 1, 3, 1, 3, 2, 3, 2, 3, 1, 3, 3, 1, 3, 3, 1, 1, 3, 1, 3, 3, 3, 1, 3, 2, 3,
3, 3, 1, 3, 3, 3, 3, 3, 1, 3, 2, 3, 3, 3, 1, 3, 1, 3, 1, 1, 3, 1, 3, 3, 3, 3, 2,
3, 1, 1, 3, 2, 3, 1, 3, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3,
3, 1, 3, 1, 3, 3, 2, 3, 3, 3, 1, 3, 2

4. Losowanie różnych elementów z populacji

Przypuśćmy, że ze zbioru N-elementowego losujemy w kolejnych jednostkach czasu po jednym elemencie, przy czym jest to losowanie ze zwracaniem. Interesuje nas czas oczekiwania na wylosowanie r różnych elementów oraz jego parametry. Niech T oznacza ten czas.

Nie jest całkiem widoczne, jak wyznaczyć rozkład T, jednak samą nadzieję matematyczną można obliczyć stosunkowo łatwo. Zauważmy w tym celu, że gdy w pewnym momencie mamy już wylosowanych n różnych elementów, to czas oczekiwania T_n na pojawienie się następnego różnego od nich elementu jest zmienną losową o rozkładzie, którego charakter jest w istocie taki sam jak rozkład czasu oczekiwania na pierwszą „6” w grze w Chińczyka – oba te rozkłady są rozkładami geometrycznymi. Zmienna losowa T_n ma rozkład:

$$P(T_n = k) = \left(\frac{n}{N}\right)^{k-1} \frac{N-n}{N}, \quad k = 1, 2, 3, \ldots$$

Możemy więc obliczyć nadzieję matematyczną i wariancję zmiennych T_n:

```
> restart:
> m := sum('(1-p)^(k-1)*p*k', 'k'=1..infinity);
```

$$m := \frac{1}{p}$$

```
> w := sum('(1-p)^(k-1)*p*k^2', 'k'=1..infinity) - m^2;
```

$$w := \frac{-p+2}{p^2} - \frac{1}{p^2}$$

```
> simplify(%);
```

$$-\frac{-1+p}{p^2}$$

```
> p := (N - n)/N;
```

$$p := \frac{N-n}{N}$$

```
> m;
```

$$\frac{N}{N-n}$$

```
> w;
```

$$\frac{N^2\,(-\frac{N-n}{N}+2)}{(N-n)^2} - \frac{N^2}{(N-n)^2}$$

```
> simplify(%);
```

$$\frac{N\,n}{(-N+n)^2}$$

Nadzieja matematyczna i wariancja zmiennej T wynoszą:

```
> Er:= (N,r) -> sum('N/(N-n)', 'n'=0..r-1);
```

$$Er := (N, r) \to \sum_{'n'=0}^{r-1} '\frac{N}{N-n}'$$

```
> Wr:= (N,r) -> sum('N*n/(N-n)^2', 'n'=0..r-1);
```

$$Wr := (N, r) \to \sum_{'n'=0}^{r-1} '\frac{N\,n}{(N-n)^2}'$$

Podstawiając za N i r konkretne wartości mamy:

```
> evalf(Er(100,30)); evalf(Wr(100,30));
```

35.45407600

6.885850949

```
> evalf(Er(200,100)); evalf(Wr(200,100));
```

138.1306861
60.37514711

```
> evalf(Er(200,190)); evalf(Wr(200,190));
```

589.8125388
3017.340055

...

Ile należy wykonać losowań ze zwracaniem, aby z populacji 200 elementowej wybrać 100 różnych elementów z prawdopodobieństwem 0,95?

Mamy znaleźć liczbę losowań x, aby $P(T < x) > 0,95$. Możemy od razu założyć, że $x > m = E(T)$. Niech $\varepsilon = x - m$. Mamy teraz kolejno:

$$P(T<x) = P(T < m+\varepsilon) = 1-P(T \geqslant m+\varepsilon) \geqslant 1-P(|T-m| \geqslant \varepsilon) \geqslant 1-\frac{D^2(T)}{\varepsilon^2}.$$

Wystarczy więc dobrać x tak, aby $1 - \frac{D^{(T)}}{(x-m)^2} \geqslant 0,95$. Ponieważ znamy już $m = 138,1306861$ oraz $D^2(T) = 60,37514711$ możemy wyliczyć x:

```
> m := 138.1306861:
  w := 60.37514711:
  solve(1 - w/(x-m)^2 > 0.95,x);
```

$$RealRange(-\infty, Open(103.3815431)),$$

$$RealRange(Open(172.8798291), \infty)$$

Tak więc przy 173 rzutach mamy 95% pewności, że wylosujemy 100 różnych elementów. Jeżeli wystarczy nam 90% pewności powinniśmy wykonać jedynie 163 rzuty.

...

Czy poprzedni wynik można polepszyć stosując centralne twierdzenie graniczne?

Z formalnego punktu widzenia, trudno stosować tutaj centralne twierdzenie graniczne, gdyż nie jest oczywiste, że w naszym przypadku są spełnione jego założenia (w najczęściej cytowanej wersji Lindeberga - Levy'ego nie są spełnione). Pytamy jednak, czy mimo tego nasza zmienna losowa T oznaczająca liczbę potrzebnych losowań, ma rozkład normalny. Sprawdzimy „normalność” zmiennej losowej T „doświadczalnie" przeprowadzając odpowiednią symulację komputerową.

Odświeżamy system i wgrywamy najpierw potrzebne moduły

```
> restart;

> with(stats):
> with(stats[statplots]): with(stats[transform]):
> with(plots):
```

Powtarzamy 500 doświadczeń polegających na wylosowaniu 100 różnych elementów z 200 elementowej populacji.

```
> nprob := 500:
> dane := NULL:
> for i from 1 to nprob do
> lista := NULL: n := 0: raz :=0:
> while nops([lista]) < 100 do
  while member(raz,[lista]) do
  los := rand(1..200):
  raz := los(): n := n+1 od;
  lista := lista,raz:
  od:
  dane := dane,n:
  od:
```

Przygotowujemy się do wykresu histogramu wraz z wykresem gęstości rozkładu normalnego o odpowiednich parametrach. Końce i szerokość przedziałów klasowych dobieramy w sposób w pewnym sensie, [1], optymalny.

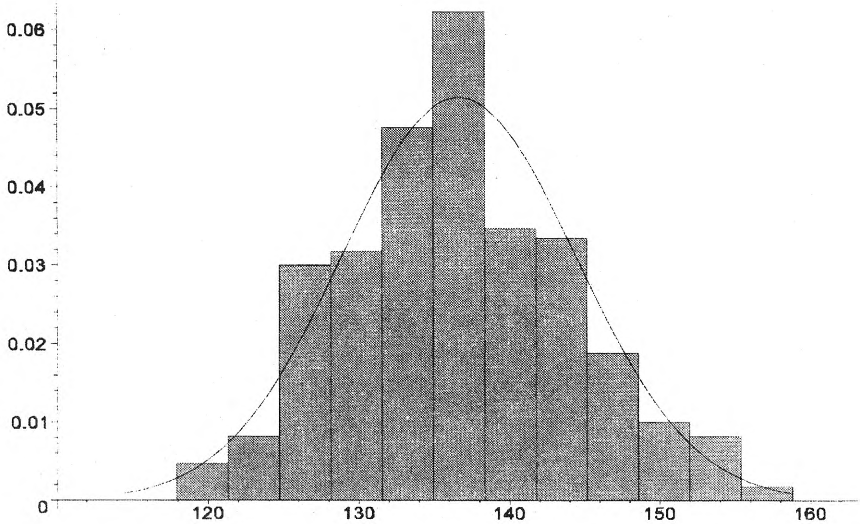
```
> m := evalf(describe[mean]([dane]));
  sigma := evalf(describe[standarddeviation]([dane]));
  c:= 3.486*sigma: h := evalf(c/nprob^(1/3));
```

$$m := 136.6480000$$
$$\sigma := 7.757325312$$
$$h := 3.407083044$$

```
> k := floor(3*sigma/h):
  a := m - (k+0.5)*h:
  b := m + (k+0.5)*h:
  dane1:=tallyinto['skrajne']([dane],[seq(a+i*h..a+(i+1)*h,
  i=0..2*k)]);
  dane2:=scaleweight[1/nops([dane])](dane1):
  hist := histogram(dane2,colour=cyan):
> pp:=plot(stats[statevalf,pdf,normald[m,sigma]],a..b,
  color=black):
> display({pp,hist});
```

```
> skrajne;
```

$$dane1 := [\text{Weight}(155.3869567..158.7940398, 3),\\ \text{Weight}(138.3515415..141.7586246, 59),\\ \text{Weight}(124.7232093..128.1302924, 51),\\ \text{Weight}(145.1657076..148.5727906, 32),\\ \text{Weight}(148.5727906..151.9798737, 17),\\ \text{Weight}(134.9444585..138.3515415, 106),\\ \text{Weight}(131.5373754..134.9444585, 81),\\ \text{Weight}(141.7586246..145.1657076, 57),\\ \text{Weight}(121.3161263..124.7232093, 14),\\ \text{Weight}(114.5019602..117.9090432, 0),\\ \text{Weight}(117.9090432..121.3161263, 8),\\ \text{Weight}(128.1302924..131.5373754, 54),\\ \text{Weight}(151.9798737..155.3869567, 14)]$$



Rysunek 6.

$$[114, 159, 159, 161]$$

Wypisaliśmy na koniec wyniki skrajne nie uwzględnione w histogramie.

Otrzymany wykres sugeruje, że zmienna losowa T ma rozkład normalny. Co więcej zauważmy, że w co najwyżej 17 doświadczeniach liczba losowań była większa niż 152, jedynie w trzech była $\geqslant 159$, a nigdy nie przekroczyła 161. Zakładając więc, że zmienna losowa T ma rozkład normalny i znając jej nadzieję matematyczną oraz wariancję — obliczone wcześniej, możemy łatwo poprawić wcześniejszy wynik. Mianowicie liczba x taka, że $P(T < x) > 0,95$ wynosi:

```
> x:=statevalf[icdf,normald[m ,sigma]](0.95);
```

$$x := 150.9114366$$

Wynik ten jest wyraźnie zgodny z przeprowadzoną symulacją.

. . .

Niech R oznacza liczbę różnych elementów otrzymanych podczas 150 losowań ze zwracaniem ze zbioru 200 elementowego. Znajdziemy graficznie rozkład R.

Postępując podobnie jak poprzednio otrzymamy odpowiednie wykresy i wartości skrajne:

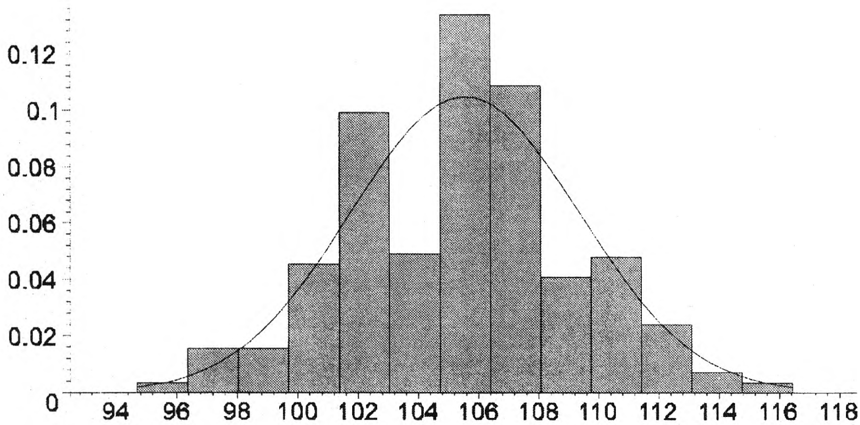
```
> nprob := 500:
> dane := NULL:
> for i from 1 to nprob do
  lista := NULL: raz :=0:
> for n from 1 to 150 do
  los := rand(1..200):
  raz := los():
  if not member(raz,[lista]) then
  lista := lista,raz fi:
  od:
  dane := dane,nops([lista]);
  od:

> m := evalf(describe[mean]([dane]));
  sigma := evalf(describe[standarddeviation]([dane]));
  c:= 3.486*sigma: h := evalf(c/nprob^(1/3));
```

$$m := 105.5580000$$
$$\sigma := 3.810857646$$
$$h := 1.673760986$$

```
> k := floor(3*sigma/h):
  a := m - (k+0.5)*h: b := m + (k+0.5)*h:
  dane1:=tallyinto['skrajne']([dane],[seq(a+i*h..a+(i+1)*h,
  i=0..2*k)]);
  dane2:=scaleweight[1/nops([dane])](dane1):
  hist := histogram(dane2, colour=cyan):
> pp:=plot(stats[statevalf,pdf,normald[m,sigma]],a..b,
  color=black):
> display({pp,hist});
> skrajne;
```

dane1 := [Weight(113.0899244..114.7636854, 6),
Weight(101.3735975..103.0473585, 83),
Weight(96.35231458..98.02607556, 13),
Weight(109.7424025..111.4161635, 40),
Weight(114.7636854..116.4374464, 3),
Weight(106.3948805..108.0686415, 91),
Weight(98.02607556..99.69983655, 13),
Weight(103.0473585..104.7211195, 41),
Weight(111.4161635..113.0899244, 20),
Weight(108.0686415..109.7424025, 34),
Weight(99.69983655..101.3735975, 38),
Weight(104.7211195..106.3948805, 112),
Weight(94.67855359..96.35231458, 3)]



Rysunek 7.

[117, 117, 118]

Literatura

[1] L. Gajek, M. Kałuszka *Wnioskowanie statystyczne, modele i metody*, Wydawnictwa Naukowo-Techniczne, 1996.

[2] *Maple V Learning Guide*, Springer, 1998.

[3] *Maple V Programming Guide*, Springer, 1998.

[4] J. Ombach, *Rachunek prawdopodobieństwa wspomagany komputerowo – Maple*, Wydawnictwo Uniwersytetu Jagiellońskiego, 2000.

[5] J. Ombach, *Wykłady z równań różniczkowych wspomagane komputerowo — Maple*, Wydawnictwo Uniwersytetu Jagiellońskiego, 1999.

[6] D. Small, J. Hosack, *Ćwiczenia z analizy matematycznej z zastosowaniem systemów obliczeń symbolicznych*, Wydawnictwa Naukowo-Techniczne, 1995.

Instytut Matematyki
Uniwersytet Jagielloński
ul. Reymonta 4
30-059 Kraków
Poland
E-mail: ombach@im.uj.edu.pl