1. Gyakorlat

Maple ismerkedés

Maple-rl általában, ajánlott maple konfiguráció, maple dokumentumtípusai, dokumentumszerkesztés, hasznos gyorsbillentyk, context menü, 1D/2D mód, szövegmez vs. kiértékelési mez, case sensitivity

Gyorsbillentyk:
F5 - váltás matematikai/szöveges beviteli mód között
Enter - kiértékelés
Ctrl+= - inline kiértékelés
Jobbra nyíl - kilépés indexbl/törtbl
Tab - placeholderek közötti navigálás
F1, Ctrl+F1, Ctrl+F2 - Maple helpek
F2 - Maple help kulcsszóhoz
Esc - szimbólum-kiegészítés
Ctrl+Del - input/output törlése
Shift+Enter - új sor kiértékelési mezben
?parancs - parancs súgója
:= - értékadás
-> - hozzárendelés, függvény, lambda
= - egyenlet

Menü:
Edit/Delete Element;Remove Output
View/Palettes;Zoom Factor;Show/Hide Contents;Collapse All Sections
Insert/Section;Subsection;Image;Table;Code Edit Region;Label
Format/Indent;Outdent;Convert To
Tools/Options;Assistants;Math Apps;Tutors
Help/Quick Reference;Manuals, Resources, and more

Menü ikonok

Paletták/Favourites

Utasítások lezárása ; vs. :, beépített help, alapvet adattípusok, szám vs. szimbolikus kifejezés, aritmetikai mveletek, értékadás, kiértékelési lánc, with, restart, függvények hívása, diff, int, solve, fsolve, isolate, plot, %, eval, evalf, cat/||, sum, unassign, op, nops, length, table, kétféle felsvessz, indexelés, programozási alapismeretek (vezérlési struktúrák)

A valószínség fogalma

A fogalom háttere

Determinisztikus és véletlenszer folyamatok

Bizonyos folyamatok kimenetele a körülmények "elég jó" ismeretében "elég jól" leírható. Pl. a megfelel fizikai törvényszerségek ismeretében meg tudjuk mondani, mi történik egy vasgolyóval, ha vákuumban, levegben, vízben stb. elengedjük - számolva a gravitáció, a közegellenállás stb. hatásaival, és a kezdeti paraméterekkel, mint pl. az ejtési magasság vagy a kezdsebesség. Ki tudjuk számítani (bizonyos pontossággal) az esés idejét, a becsapódás sebességét stb.
Az ilyenek az ún. *determinisztikus* (jól meghatározott) folyamatok. Az ilyen jelleg folyamatoknál érdemes törekedni a mködést befolyásoló tényezk mind pontosabb megismerésére, a kezdeti feltételek minél pontosabb meghatározására, a számítási eljárások tökéletesítésére, mert ezekkel van remény a folyamat egyre pontosabb leírására, és végállapot meghatározására. Jól meghatározott, kiszámítható folyamat során juthat el pl. egy reszköz egy távoli bolygóra.
Ezzel szemben vannak olyan jelenségek is, melyek kimenetele nem jósolható adott szinten hatékonyan, mert pl. olyan bonyolult törvényszerségek írják le és/vagy olyan pontatlanul mérhetk a kezdeti feltételek, hogy egy adott kísérlet lefolyására és végállapotára nem lehet érdemi (elfogadható hibán belüli) elrejelzést adni. Az ilyeneket szokták a determinisztikus folyamatok ellenpontjaként véletlenszer jelenségnek nevezni. Pl. gyakorlatilag teljesen véletlenszernek tekinthetjük a pénzérmével vagy kockával való dobás végeredményét - hogy csak a legegyszerbbeket említsük.
Filozófiai értelemben persze nincs éles határ a determinisztikus és a véletlenszer folyamatok között.
Egy golyó leesésének vizsgálata elméletileg viszonylag egyszer, de az esés idejére vonatkozó elrejelzésnek is hibája van, mivel a kísérlet, valamint a mérés sem végezhet el kétszer egyformán. Korrekten soha nem mondhatjuk azt, hogy az adott kísérleti összeállításban golyó esésének ideje pl. éppen 3 másodperc volt, hanem csak annyit, hogy pl. 0.1 vagy 0.01 másodperc pontossággal annyi. Vagyis itt is megjelenik a véletlen, de a bizonytalanság az adott szinten elfogadható mértéken belül marad, ráadásul a kísérlet egyre jobb kivitelezésével csökkenthet, és elvileg ez cél is.
Ezzel szemben bár a pénzérme feldobásakor is jól ismert fizikai folyamatok mködnek, ezek annyira összetettek, és annyira függnek a kezdeti feltételektl (eldobás sebessége, szöge, perdület, esési magasság), hogy hétköznapi körülmények között gyakorlatilag nem reprodukálható kétszer egyformán a kísérlet. Az érme pörögve esése során a két kimenetel (fej-írás) lehetsége olyan gyorsan változik, hogy a repülés jósolható idejének hibáján belül is mindkett többször elfordul, ezért gyakorlatilag véletlenszernek tekinthet. Valójában lemondunk arról, hogy *egy adott* kísérlet kimenetelére érdemi (értelmes hibahatáron belüli) elrejelzést tudjunk adni, lemondunk a folyamat részleteinek a vizsgálatáról. Erre azonban ne feltétlenül valamiféle kényszer megalkuvásként tekintsünk, hanem mint a világ izgalmas sokszínségének egyik alapvet megnyilvánulására, ami nem is kezelhetetlen, csak megfelel szemléletmódot és eszközöket igényel.

Valószínségszámítás és a statisztika

Ha a fentiek szerint lemondunk arról, hogy *egyetlen adott* kísérlet eredményére determinisztikus úton adott szinten elfogadható elrejelzést adjunk, ez még nem jelenti, hogy teljesen eszköztelenek lennénk. El kell fogadnunk, hogy a világ túl bonyolult ahhoz, hogy mindig minden kérdésre egyszer és pontos legyen a válasz, és hogy a lokálisan bizonytalannak tn események mögött is lehet rendszer.
Filozófiailag érdekes kérdés lehet általánosítva, hogy egy ilyen rendszer megismerhet-e mindig a tudatunkkal, ill. ha nem, akkor nevezhetjük-e egyáltalán rendszernek (vagy ez már

az, amit inkább "Isten"-nek szokás). Mindenesetre a valószínségszámításban egy még matematikai eszközökkel tárgyalható szép példát találunk a lokális bizonytalanságok mögött megbújó rendszerre.
A valószínségszámítás során lemondunk arról, hogy egy adott esemény bekövetkezésére konkrét jóslatot adjunk, mivel elfogadjuk, hogy többféleképp is bekövetkezhet. Elképzeljük viszont, hogy ha *azonos körülmények mellett sokszor elvégeznénk* ugyanazt a kísérletet, akkor az esetek *milyen arányában* következne be egy adott kimenet (esemény). Pl. a pénzfeldobás esetén értjük, hogy *egy adott* kísérletnél nem lehet tudni, fej lesz-e vagy írás (márpedig csakis egyik), ugyanakkor azt is érezzük, nincs okunk feltételezni, hogy bármelyikük bekövetkezése mellett érvek szólnának a másik rovására; ebbl azt gondoljuk, hogy ha *sokszor* dobjuk fel a pénzt, akkor a kimenetek között fej és írás is szerepelni fog, nagyjából egyenl (50-50%) arányban. Ez az, amit úgy nevezünk, hogy pl. a "fej" dobás valószínsége 0.5 (50%).
Valójában a nagyon sok (elméletileg végtelenbe tartó számú), azonos körülményekkel elképzelt kísérleteken belüli relatív gyakoriság (határértéke) tehát a valószínség. Szoktuk azt is mondani, hogy egy érme feldobásakor a "fej" valószínsége 50%, de ennek közvetlen jelentése egyetlen kísérletnél valójában nincs, csak az elépzelt sok kísérlet során megmutatkozó arány vetül vissza ebben a megfogalmazásban. Ugyanakkor gyakorlati szempontból egyetlen (vagy nem túl sok) kísérletnél is lehet értelme a valószínség ismeretének, mert segítheti a döntést. Ha pl. valaki ajánl egy egyszeri fogadást, hogy pénzfeldobáskor "fej" esetén fizet pl. 1000 forintot, de "írás" esetén nekünk csak 100-at kell, eldönthetjük, érdemes-e megpróbálnunk.
Vegyük észre, hogy ezzel az egyszer példával modellezett helyzettekkel valójában nap mint nap találkozunk, életünk során folyamatosan bizonytalan kimenetel eseményekkel kapcsolatban kell döntéseket hoznunk a kockázatok és a várható nyereségek elemzésével (természetesen nem csak számszersíthet dolgokról van szó). Eközben a begyjtött információk során folyamatosan kalibráljuk a fejünkben a dolgokhoz tartozó valószínségeket, eldöntve, mit érdemes bevállani, mit nem. Hosszabb távon az nyer, aki a valószínségeket jól ítéli meg, és azokhoz mérten hozza a döntéseit. Aki ezt jól csinálja, azt szokta a közvélemény "tapasztalt"-nak tartani. Ezt a képességet trenírozzák (persze speciális helyzetekre) a különféle szerencsejátékok is - pl. a jó pókerjátékost is gyenge lapjárásnál egy-egy alkalommal bármelyik kezd megverheti, de hosszabb távon a tapasztalt játékos mégis nyer, mert jobban méri fel a helyzeteket és a hozzájuk tartozó valószínségeket - tudja, mikor érdemes többet kockáztatni, és mikor kell veszni hagyni esetleg kevesebbet.
Azt szokták mondani, hogy a valószínségszámítás valami elméleti konstrukcióból próbál következtetni a valószínségre, ebbl pedig egy kísérletsorozat során megfigyelhet relatív gyakoriságokra. Pl. amikor fentebbi meggondolásaink alapján azt mondtuk, hogy érme feldobásakor a "fej" valószínsége 50%, ezzel azt állítjuk, hogy egy sokdobásos kísérletsorozatban az esetek kb. felében lesz fej az eredmény.
Ehhez képest a statisztika egy konkrét véges mintából próbál következtetni a háttérben húzódó valószínségi modell paraméthereire. Pl. ha egy 1000 dobásos kísérletsorozatban 619-szer jön ki a "fej", akkor vizsgálhatjuk, mennyire hihet az adott érmére az 50-50% megoszlás, van-e valami szignifikáns tényez, ami ezt torzítja, vagy az eredmény belefér a véletlenszer ingadozásba. Kiderülhet a vizsgálatból, hogy más valószínségi modellt kell használni.

1. Pénzfeldobás szimulációja

A valószínségszámítás során tömegjelenségeket vizsgálunk, egy-egy kísérlet sokszori elvégzését képzeljük el. A számítógépes környezet rendkívül alkalmas ezek szimulációjára, vizsgálatára. A matematikai tartalom mellett párhuzamosan a Maple rendszer kapcsolódó részeit is áttekintjük.

Bevezet példaként a pénzfeldobás egyszer esetén tekintjük át az alapokat.

Mindenek eltt javasolt alaphelyzetbe állítani a program állapotát (az utasításokat gyakran pontosvesszvel zárjuk):

```
> restart;
```

Pénzfeldobás, elképzelt szimuláció

Adjuk meg egy egydimenziós listában egy tíz kísérletbl álló dobás-sorozat elképzelt lefolyásának eredményét:

```
> elkepzelt_dobasok := [F,I,I,F,I,F,F,I,F,I];
```

Ez egy programozásból is ismert tömbként viselkedik. Pl. megkérdezhetjük, hány elem a lista:

```
> nops(elkepzelt_dobasok);
```

Rá is kérdezhetünk bármelyik elemére. (Az indexelés 1-tl kezddik!)

```
> elkepzelt_dobasok[1], elkepzelt_dobasok[2],
  elkepzelt_dobasok[3];
```

A nem létez indexek lekérdezésekor hibát kapunk:

```
> elkepzelt_dobasok[0];
> elkepzelt_dobasok[11];
```

Pénzfeldobás valódi érmével

Vegyünk el egy valódi pénzérmét, hajtsunk végre egy valódi 10 elem dobás-sorozatot és rögzítsük az eredményt egy listában:

```
> dobasok := [F,F,F,I,F,F,I,F,I,F];
```

Ellenrizzük a dobások számát:

```
> nops(dobasok);
```

Pénzfeldobás szimuláció a Maple segítégével

```
> restart;
```

A szükséges apparátus a *Statistics* csomagban van, elbb ezt be kell hívni. Olyan ez, mint pl. amikor Java-ban küls osztályokat importálunk.

```
> with(Statistics):
```

Az utasítást ezúttal kettsponttal zártuk le. Próbáljuk ki, mi történik, ha pontosvesszvel zárjuk (majd ENTER). (Felsorolja a csomagban elérhet eljárásokat.)

Létrehozunk egy valószínségi változót. Ez a Maple-ben több, mint egy egyszer véletlenszám-generátor, de annak is használható. Olyanra van szükségünk, ami két értéket állít el: pl. 0, 1.

```
> Y := DiscreteUniform(0, 1);
```

Ezzel generálhatunk valahány elem mintát. Rögzítsük hozzá a darabszámot egy változóban, hogy azon keresztül egységesen hivatkozhassuk (és módosíthassuk) a folyamat különböz pontjain.

```
> darab:=10;
> minta := Sample(Y, darab);
```

Ha a fentieket ('restart' paranccsal együtt) újra és újra lefuttatjuk (pl. végig ENTER-ezzük), mindig ugyanazt az eredményt kapjuk - ha 'restart' nélkül (akár csak magát a mintagenerálást), akkor mindig mást. Ennek az a magyarázata, hogy a véletlenszám-generátor valójában egy determinisztikus matematikai algoritmus szerint mködik, mely adott (a restart által alapra

állított) kezdértékbl indítva mindig ugyanazt a számsort generálja.
Ha ténylegesen véletlenszervé akarjuk tenni a generált számokat, akkor a kezdértékét inicializálni kell:

```
> randomize();
```

Ez az utasítás a bels órából kiolvasott értékkel inicializálja a véletlenszám-generátort, így az ismételt futásokkor már ténylegesen más és más, véletlenszer értékek keletkeznek:

```
> minta := Sample(Y, darab);
```

A 0,1 értékek megfeleltethetk más (pl. *F, I*) szimbólumoknak. Vegyük fel egy listába a kívánt kimeneti értékkészletet:

```
> ertekek := [F, I];
```

Egyszer szekvenciával átírhatjuk a mintát az új értékkészletre:

```
> generalt_dobasok := [seq(ertekek[round(minta[i] + 1)], i=
  1..darab)];
```

Ezek után ez természetesen nagyobb darabszámú mintára is hasonlóan elkészíthet - akár a parancsok egyetlen blokkba összemásolásával:

```
> darab := 100;
  minta := Sample(Y, darab);
  generalt_dobasok := [seq(ertekek[round(minta[i] + 1)], i=
  1..darab)];
```

Érdekes megfigyelni, hogy a valóságos folyamatban (valódi pénzfeldobással, ill. a valódi helyzetet helyesen szimuláló Maple segítségével generált sorozatban) könnyen elfordulhatnak olyan 4-5-szöri ismétldések, amiktl a találomra, fejbl generált sorozatnál tartózkodtunk. Ez számításokkal is alátámasztható, de itt most nem cél ennek részletezése.

A "fej" dobás relatív gyakoriságának alakulása egy kísérletsorozatban

Generáljunk egy dobás-sorozatot az elzek szerint. Elször egy kisebb darabszámmal nézzük, hogy jobban érthetek legyenek a lépések. (Shift+ENTER végrehajtás nélkül törhetjük az utasításblokk sorait, hogy áttekinthetbb kódot kapjunk.)

```
> darab := 10:
  minta := Sample(Y, darab):
  generalt_dobasok := [seq(ertekek[round(minta[i] + 1)], i=
  1..darab)];
```

Számoljuk össze, hogy az *i.* kísérletig hányszor lett fej, és ezt az adatot (gyakoriság) rögzítsük egy tömbben.

```
> fej := 0:
  for i from 1 to darab do
     if generalt_dobasok[i]=F then
        fej:=fej+1 end if: gyakorisagok[i]:=fej
  end do;
```

Az elfordulások darabszáma természetesen kumulatív jelleggel monoton növekszik - igazi információértéke az elfordulási aránynak van az összes elvégzett kísérlet között.
Ezen értékeknek (gyakoriságoknak) az addigi kísérletek számához mért aránya a relatív gyakoriság. Ennek alakulását rögzítsük egy listában:

```
> relativ_gyakorisagok := [seq(gyakorisagok[i] / i, i = 1..
  darab)];
```

Rendezzük az adatokat sorszámaikkal vett párokba, hogy ábrázolható pontok koordinátáit kapjuk:

```
> pontok := [seq([i, relativ_gyakorisagok[i]], i = 1..darab)
  ];
```
Ezeket mint koordináta-párok listáját pontként ábrázolhatjuk diagramon:
```
> plot(pontok, style=point);
```
Ha több dologból szeretnénk rajzolni egy ábrát, akkor az egyes részábrákat külön változókban kell tárolni, majd a 'plots' csomag 'display' parancsával egy rajzban összemásolni ket. A pontok mellé rajzolunk egy vonalat is $\frac{1}{2}$ (a fej dobás valószínsége) magasságában:
```
> pontok_plot := plot(pontok, style=point):
  vonal_plot := plot(1/2, x = 0..darab, y=0..1, color=green)
  : vonal_plot:
  plots[display](pontok_plot, vonal_plot);
```
A relatív gyakoriság a kísérletek végén (a kevés számú kísérlet miatt a valószínség nem feltétlenül túl jó közelítéseként):
```
> relativ_gyakorisagok[darab]; evalf(%);
```
Foglaljuk össze az eddigi folyamat utasításait egyetlen utasításblokkban (Shift+ENTER sortörésekkel) - a felesleges köztes eredmények kiírása nélkül (kettspontra cserélve a pontosvesszket). Csupán az egyesített ábra és a végleges relatív gyakoriság jelenjen meg az eredményben - ezek mögött hagyunk pontosvesszt:
```
> Y := DiscreteUniform(0, 1):
  darab := 1000:
  minta := Sample(Y, darab):
  generalt_dobasok := [seq(ertekek[round(minta[i] + 1)], i=
  1..darab)]:
  fej := 0:
  for i from 1 to darab do
    if generalt_dobasok[i]=F then
      fej:=fej+1
    end if:
    gyakorisagok[i]:=fej
  end do:
  relativ_gyakorisagok := [seq(gyakorisagok[i] / i, i = 1..
  darab)]:
  pontok := [seq([i, relativ_gyakorisagok[i]], i = 1..darab)
  ]:
  pontok_plot := plot(pontok, style=point):
  vonal_plot := plot(1/2, x = 0..darab, y=0..1, color=green)
  : vonal_plot:
  # gyok_plot := plot({1/2+1/(2*sqrt(x)), 1/2-1/(2*sqrt(x))}
  , x = 0..darab, y = 0..1, color=black): gyok_plot:
  # plots[display](pontok_plot, vonal_plot, gyok_plot);
  plots[display](pontok_plot, vonal_plot);
  relativ_gyakorisagok[darab]; evalf(%);
```

E fenti blokk újra és újra futtatásával jól láthatóan más és más (véletlenszer), de jellegében az elméleti valószínséghez konvergáló értékeket kapunk.
A darabszámot mint paramétert változtatva jól látszik, hogy a közelítés egyre magasabb darabszámokkal egyre jobb lesz.

2. Két pénzérmét dobunk fel egyszerre vagy egy pénzérmét dobunk fel kétszer. Eseménytér és az elemi események valószínségei.

2. FELADAT
Adjuk meg az eseményteret, ha a kísérlet során egy pénzérmét dobunk fel kétszer!
Adjuk meg az eseményteret, ha a kísérlet során két pénzérmét dobunk fel egyszer!
Milyen valószínségeket kell adnunk az elemi eseményeknek, ha az érme szabályos?
Vonjunk le következtetést!

Megoldás:

1. modell. Egy pénzérmét dobunk fel kétszer.

Meg tudjuk különböztetni a dobás sorrendjét!

```
> restart;
> alap := [0, 1];
> M[1] := Matrix(2, 2, (elso, masodik) -> [alap[elso], alap
  [masodik]]);
```

Szabályos esetben $P((0,0)) = P((1,0)) = P((0,1)) = P((1,1)) = \frac{1}{4}$

```
> Omega[1] := Matrix(2, 4, [convert(M[1], list), [seq(1/4, k =
  1..4)]]);
```

2. modell. Kett pénzérmét dobunk fel egyszerre.

Nem tudjuk a sorrendet megkülönböztetni!

```
> M[2] := {[0, 0], [0, 1], [1, 1]};
```

$P((0,0)) = P((1,1)) = \frac{1}{4}$, $P((1,0)) = \frac{1}{2}$.

```
> Omega[2] := Matrix(2, 3, [convert(M[2], list), [1/4, 1/2,
  1/4]]);
```

Következtetés.
Használjuk inkább az **1. modell** eseményterét, mert ott minden kimenetel egyformán valószín!

3. Kocka dobás kétszer. Eseménytér, események, mveletek eseményekkel. Valószínségek.

3. FELADAT
Két szabályos 6 oldalú dobókockával dobunk. Jelölje A azt az eseményt, hogy az **összeg 4, 5 vagy 6** lesz, míg B azt, hogy **mindkét dobás páros**
$A = \{ (a, b) \mid a + b = 4 \text{ vagy } 5 \text{ vagy } 6, \text{ ahol } a, b \in \{1, 2, 3, 4, 5, 6\} \}$, $B = \{ (a, b) \mid a, b \in \{2, 4, 6\} \}$
(i) Adja meg az Ω eseménytér elemeit Maple segítségével!
(ii) Adja meg az A és B eseményeket halmazokkal! Számolja ki a $P(A)$ és $P(B)$ valószínségeket!
(ii) Határozza meg az $A \cap B$ **együttes eseményt** szavakkal és adja meg az elemeit Maple-ben!

MEGOLDÁS
(i) Két kocka dobásnál az összes variációt egy 6×6 méret mátrixban célszer megadni!

```
> restart;
  Omega := Matrix(6, 6, (i, j) -> [i, j]);
```

Itt az Ω mátrix 6 sorból és 6 oszlopból áll, azaz egy 6×6 *méretű* mátrix. A mátrix i-edik sor j-edik oszlopában az $[i, j]$ lista áll, amely azt jelzi, hogy az els dobás értéke i és a második dobás értéke j lett és $i, j \in \{1, 2, 3, 4, 5, 6\}$ lehet. Ebbl könnyen kiolvasható, hogy az összes párosításra $6 \cdot 6 = 36$ lehetség adódik. Az eseménytér, mint halmaz a fenti mátrixból halmaz konverzióval a következképpen kapható meg a Maple-ben.

```
> eseménytér := convert(Omega, set);
  N := nops(eseménytér);
```

(ii) Adja meg az A és B eseményeket halmazokkal!

Az $A = \{ [i,j] | i+j = 4, 5, 6 \}$ esemény megadásához készítünk egy olyan 6×6-os mátrixot, melyben a megfelel sor és oszlop indexek összegei szerepelnek. Az így kapott mátrixból ki kell gyjteni az összes 4, 5 *és* 6 számokhoz tartozó sor és oszlop indexet!

```
> összegek = Matrix(6, 6, (i, j) -> i + j);
```

```
> A := {[1, 3], [1, 4], [1, 5], [2, 2], [2, 3], [2, 4], [3, 1],
  [3, 2], [3, 3], [4, 1], [4, 2], [5, 1]};
  nA := nops(A);
  PA := nA/N;
```

Látható, hogy az A esemény $nA = 12 -$ féleképpen következhet be. Mivel minden kimenetel egyformán valószín, ezért $P(A) = \frac{12}{36} = \frac{1}{3}$

A $B = \{mindkét\ dobás\ páros\} = \{ [i,j] |\ i,j \in \{2, 4, 6\} \}$ esemény Maple megadásához használjunk két egymásba ágyazott sorozat képz *seq* eljárást, melyben a lépték 2.

```
> B := {seq(seq([i, j], j = 2 .. 6, 2), i = 2 .. 6, 2)};
  nB := nops(B);
  PB := nB/N;
```

A küls sorozattal az els dobás i értékét állítjuk el, mely 2 és 6 között változik 2 lépésközzel. A bels sorozat a második dobás j értékét változtatja hasonló módon. Így a B halmaz elemeinek száma $nB = 3 \cdot 3 = 9$. Mivel minden kimenetel egyformán valószín, ezért $P(B) = \frac{9}{36} = \frac{1}{4}$

(iii) Határozza meg az $A \cap B$ együttes eseményt szavakkal és adja meg elemeit Maple-ben!

Az $A \cap B$ eseményhez tartozó kimenetelek mind az A és mind a B események "*tulajdonságait*" örökli. Tehát

$$A \cap B = \{ [i,j] \mid az\ (i+j)\ összeg\ 4,\ 5\ vagy\ 6\ és\ i,j\ mindegyike\ páros \}.$$

Az A és B események együttesen úgy következhetnek be, ha a két dobás összege 4, 5 vagy 6, valamint mindegyik dobás páros. Mivel páros számok összege csak páros lehet, ezért az összeg nem lehet 5. Így a B halmaz elemei közül ki kell válogatni azokat, amelyek összege 4 vagy 6. Hasonlóan, az A halmaz elemei közül ki kell válogatni azokat, amelyek mindkét eleme páros. Ezt egyszeren megteszi nekünk a halmazok metszetére beépített **intersect** Maple eljárás.

```
> `A metszet B` := A intersect B;
  nAB := nops(%);
  PAB := nAB/N;
```

Tehát az $A \cap B$ metszet halmaznak 3 eleme van. Mivel minden kimenetel egyformán valószín, ezért $P(A \cap B) = \frac{3}{36} = \frac{1}{12}$.

Az alábbi (a)-(c) ábrákon rendre szürke színnel kiemeltük az A, a B események, valamint az $A \cap B$ metszet halmaz elemeit az eseménytér 36 eleme közül.

[1, 1]	[1, 2]	[1, 3]	[1, 4]	[1, 5]	[1, 6]
[2, 1]	[2, 2]	[2, 3]	[2, 4]	[2, 5]	[2, 6]
[3, 1]	[3, 2]	[3, 3]	[3, 4]	[3, 5]	[3, 6]
[4, 1]	[4, 2]	[4, 3]	[4, 4]	[4, 5]	[4, 6]
[5, 1]	[5, 2]	[5, 3]	[5, 4]	[5, 5]	[5, 6]
[6, 1]	[6, 2]	[6, 3]	[6, 4]	[6, 5]	[6, 6]

[1, 1]	[1, 2]	[1, 3]	[1, 4]	[1, 5]	[1, 6]
[2, 1]	[2, 2]	[2, 3]	[2, 4]	[2, 5]	[2, 6]
[3, 1]	[3, 2]	[3, 3]	[3, 4]	[3, 5]	[3, 6]
[4, 1]	[4, 2]	[4, 3]	[4, 4]	[4, 5]	[4, 6]
[5, 1]	[5, 2]	[5, 3]	[5, 4]	[5, 5]	[5, 6]
[6, 1]	[6, 2]	[6, 3]	[6, 4]	[6, 5]	[6, 6]

[1, 1]	[1, 2]	[1, 3]	[1, 4]	[1, 5]	[1, 6]
[2, 1]	[2, 2]	[2, 3]	[2, 4]	[2, 5]	[2, 6]
[3, 1]	[3, 2]	[3, 3]	[3, 4]	[3, 5]	[3, 6]
[4, 1]	[4, 2]	[4, 3]	[4, 4]	[4, 5]	[4, 6]
[5, 1]	[5, 2]	[5, 3]	[5, 4]	[5, 5]	[5, 6]
[6, 1]	[6, 2]	[6, 3]	[6, 4]	[6, 5]	[6, 6]

(a) az A halmaz elemei, (b) a B halmaz elemei, (c) az $A \cap B$ halmaz elemei

Gyakorló feladatok

3. feladat folytatása

(i) Határozza meg az $A - B$ különbség eseménynek megfelel halmazt! Adja meg szavakkal a különbség halmazt!
(ii) Számolja ki a $P(A - B)$ különbség esemény valószínségét!
(iii) Mutassa meg, hogy teljesül a $P(A - B) = P(A) - P(A \cap B)$ azonosság!
(iv) Írjon eljárást, mely TRUE vagy FALSE értéket ad vissza, attól függen, hogy a két kockadobás eredménye az A halmazba esik-e!
Ez alapján mutassa meg szimulációs programmal, hogy a $P(A)$ valószínség számítása helyes volt! Használja a kapott eljárásokat!

Kockadobás modellezése a Maple eszközeivel

a) Készítsen egy valószínségi változót (véletlenszám-generátort) a kockadobás szimulálására!
b) Generáljon 10, 100 és 1000 elem mintákat véletlenszer kockadobás-kísérletek eredményébl!
c) Készítse el a legnagyobb mintára a 6-os dobások relatív gyakoriságának alakulását mutató grafikont, és a hozzá szükséges adatokat!
d) Foglalja össze a folyamatot egyetlen utasításblokkban, melyben a minta elemszámát változóban tárolva könnyen generálható a grafikon különböz méret mintákra!