10. Gyakorlat

Maple ismerkedés

A Statistics csomag: különböz eloszlások, valószínségi változó, mintavétel (Sample), Probability, ProbabilityFunction, PDF, CDF (!), eloszlás készítése eloszlás/srségfüggvénybl, ExpectedValue, Variance, StandardDeviation, Moment, CentralMoment, ProbabilityTable, EmpiricalDistribution LineChart, Histogram+range, ColumnGraph, DensityPlot+range

```
> with(Statistics):
> X1 := RandomVariable(DiscreteUniform(1, 6));
```

$$X1 := \_R \qquad (1.1)$$

```
> Y := RandomVariable(DiscreteUniform(1, 6));
```

$$Y := \_R0 \qquad (1.2)$$

```
> X1 + Y; # 2 kockával való dobás
```

$$\_R + \_R0 \qquad (1.3)$$

```
> X2 := RandomVariable(Bernoulli(0.1)); # p
```

$$X2 := \_R1 \qquad (1.4)$$

```
> X2 := RandomVariable(Poisson(1)); # lambda
```

$$X2 := \_R2 \qquad (1.5)$$

```
> X3 := RandomVariable(Binomial(10, 0.2)); # n, p
```

$$X3 := \_R3 \qquad (1.6)$$

```
> X4 := RandomVariable(Geometric(0.2)); # p !!!Maple máshogy
  definiálja!!!
```

$$X4 := \_R4 \qquad (1.7)$$

```
> ProbabilityFunction(X4, 0);
```

$$0.20 \qquad (1.8)$$

```
> X5 := RandomVariable(Hypergeometric(100, 20, 10)); # N, S, n
```

$$X5 := \_R5 \qquad (1.9)$$

```
> X6 := RandomVariable(Uniform(0, 5)); # a, b (geometriai
  valószínség)
```

$$X6 := \_R6 \qquad (1.10)$$

```
> X7 := RandomVariable(Exponential(0.1)); # lambda=10 !!!Maple
  máshogy definiálja!!!
```

$$X7 := \_R7 \qquad (1.11)$$

```
> X8 := RandomVariable(Normal(173, 8)); # mu, sigma
```

$$X8 := \_R8 \qquad (1.12)$$

```
> ?EmpiricalDistribution
  #(<>, 'probabilities' = [])
> Probability(X8 >= 170); evalf(%);
```

$$\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{3}{16}\sqrt{2}\right)$$

$$0.6461697666 \qquad (1.13)$$

```
> Probability({X8 >= 170, X8 < 180}); evalf(%);
```

$$\frac{1}{2}\,\mathrm{erf}\left(\frac{7}{16}\sqrt{2}\right) + \frac{1}{2}\,\mathrm{erf}\left(\frac{3}{16}\sqrt{2}\right)$$

(1.14)

$$0.4553828137 \quad (1.14)$$

```
> Probability(X8 < 170) + Probability(X8 >= 180); evalf(%);
```

$$1-\frac{1}{2}\,\mathrm{erf}\left(\frac{3}{16}\sqrt{2}\right)-\frac{1}{2}\,\mathrm{erf}\left(\frac{7}{16}\sqrt{2}\right)$$

$$0.5446171863 \quad (1.15)$$

```
> [seq(ProbabilityFunction(X3, k), k = 0..10)];
```

$[0.1073741824, 0.2684354560, 0.3019898880, 0.2013265920, 0.0880803840, 0.0264241152, 0.0055050240, 0.0007864320, 0.0000737280, 0.0000040960, 1.0240\ 10^{-7}]$ (1.16)

```
> binomialis := k -> ProbabilityFunction(X3, k);
```

$$binomialis := k \rightarrow Statistics{:-}ProbabilityFunction(X3, k) \quad (1.17)$$

```
> PDF(X7, 0.2); # srségfüggvény értékei
```

$$1.353352832 \quad (1.18)$$

```
> f := x -> PDF(X7, x); # srségfüggvény (a PDF csak egy
  algebrai kifejezés)
```

$$f := x \rightarrow Statistics{:-}PDF(X7, x) \quad (1.19)$$

```
> CDF(X1, 5); evalf(%); # eloszlásfüggvény értékei, !!!Maple
  máshogy definiálja!!! - P(X <= x)
```

$$\frac{5}{6}$$

$$0.8333333333 \quad (1.20)$$

```
> CDF(X8, 0.5); CDF(X8, 10);
```

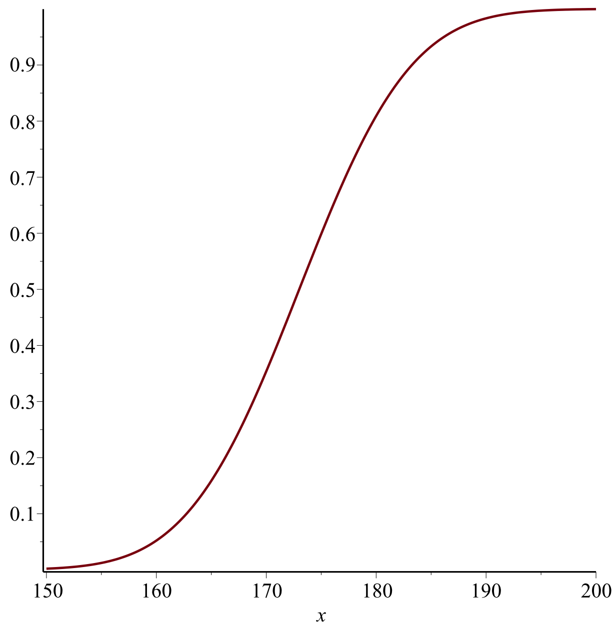
$$2.02086949288466\ 10^{-103}$$

$$\frac{1}{2}-\frac{1}{2}\,\mathrm{erf}\left(\frac{163}{16}\sqrt{2}\right) \quad (1.21)$$

```
> F := x -> CDF(X8, x);
```

$$F := x \rightarrow Statistics{:-}CDF(X8, x) \quad (1.22)$$

```
> plot(F(x), x = 150..200);
```




```
> op(Sample(X2, 20));
```

$$20, \{3 = 1., 4 = 3., 5 = 1., 7 = 3., 8 = 2., 11 = 1., 20 = 1.\}, datatype = float_8, storage = rectangular, order = Fortran\_order, shape = [\ ] \tag{1.23}$$

```
> f2 := x -> piecewise(x <= 1, 0, 1/(x^2)); # -1/x a primív
  függvénye
```

$$f2 := x \rightarrow piecewise\left(x \leq 1, 0, \frac{1}{x^2}\right) \tag{1.24}$$

```
> int(f2(x), x = 1..infinity);
```

$$1 \tag{1.25}$$

```
> int(f2(x), x = -infinity..t);
```

$$\begin{cases} 0 & t \leq 1 \\ \frac{t-1}{t} & 1 < t \end{cases} \tag{1.26}$$

```
> X20 := RandomVariable(Distribution(PDF = f2));
```

$$X20 := \_R9 \tag{1.27}$$

```
> Probability(X20 < 3);
```

$$\tag{1.28}$$

$$\frac{2}{3} \tag{1.28}$$

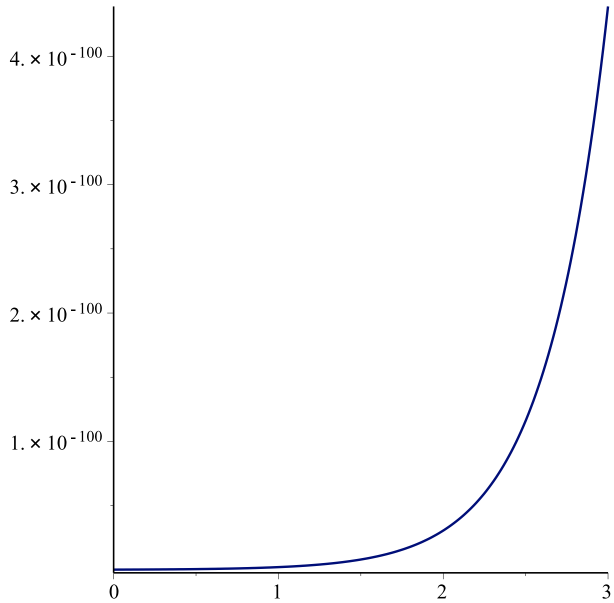
```
> X21 := RandomVariable(Distribution(CDF = (x -> piecewise(x <=0,
  0, (x-1)/x))));
```

$$X21 := \_R10 \tag{1.29}$$

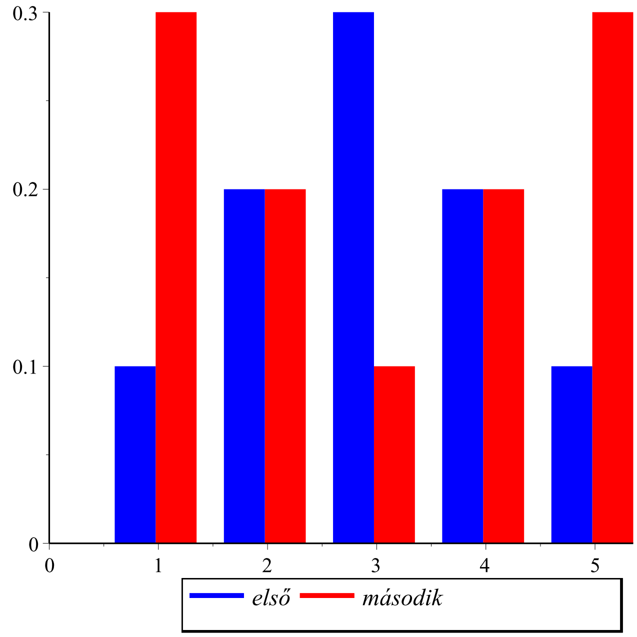
```
> Moment(X6, 2);
```

$$\frac{25}{3} \tag{1.30}$$

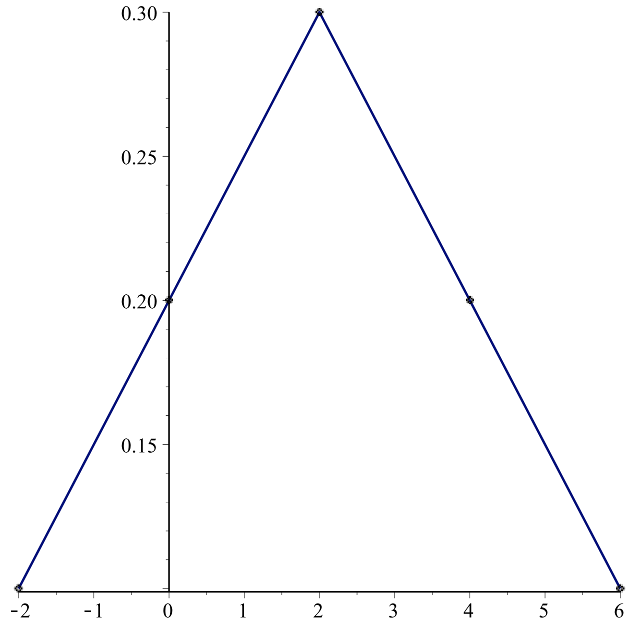
```
> DensityPlot(X8, range = 0..3); # srségfüggvény gyors
  ábrázolása
```




```
> ColumnGraph([[0.1, 0.2, 0.3, 0.2, 0.1], [0.3, 0.2, 0.1, 0.2,
  0.3]], color = [blue, red], legend = ['els', 'második'],
  offset = 0.6);
```



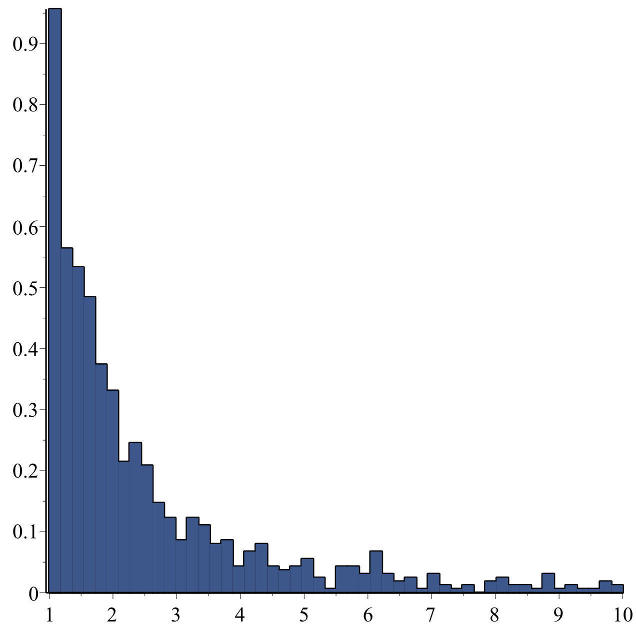

```
> LineChart([0.1, 0.2, 0.3, 0.2, 0.1], xcoords = [-2, 0, 2, 4, 6]
  ); # a range opció itt nem mködik!
```



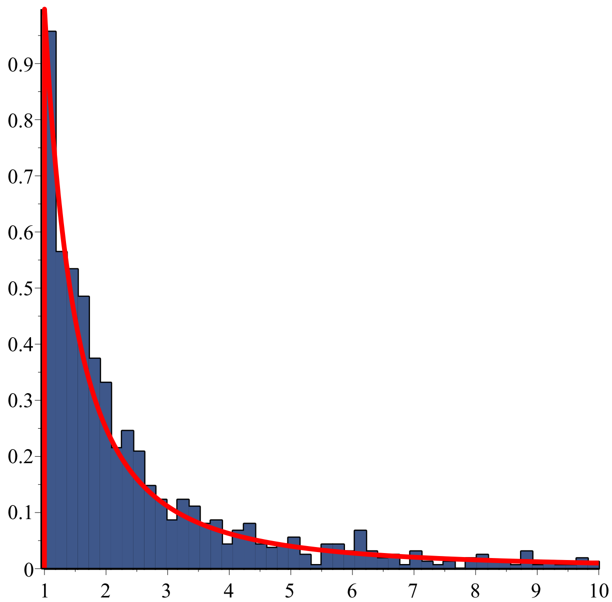

```
> minta := Sample(X20, 1000);
```

$$minta := \begin{bmatrix} \text{1 .. 1000 } Vector_{row} \\ \text{Data Type: } float_8 \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \tag{1.31}$$

```
> H := Histogram(minta, range = 1..10, bincount = 50): H;
```




```
> plots[display](H, DensityPlot(X20, range = 1..10, thickness =
  3, color = red)); # srségfüggvény illeszkedik a hisztogramra
```



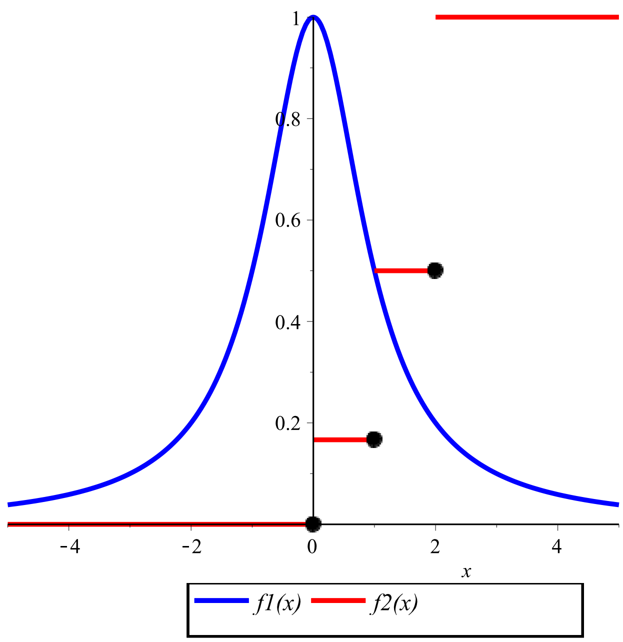
Integrálás, deriválás, határérték: diff/Diff, int/Int, limit/Limit

```
> f1 := x -> 1/(1 + x^2);
  f2 := x -> piecewise(x <= 0, 0, x <= 1, 1/6, x <= 2, 3/6, 1):
  `f2(x)` = f2(x);
```

$$f1 := x \rightarrow \frac{1}{x^2+1}$$

$$f2(x) = \begin{cases} 0 & x \leq 0 \\ \frac{1}{6} & x \leq 1 \\ \frac{1}{2} & x \leq 2 \\ 1 & otherwise \end{cases} \qquad (1.32)$$

```
> plot([f1(x), f2(x)], x = -5..5, color = [blue, red], thickness
  = 3, symbol = solidcircle, symbolsize = 20, discont = true,
  legend = [`f1(x)`, `f2(x)`]);
```




```
> diff(f1(x), x);
```

$$-\frac{2x}{(x^2+1)^2} \tag{1.33}$$

```
> diff(f1(x), x$2);
```

$$\frac{8x^2}{(x^2+1)^3} - \frac{2}{(x^2+1)^2} \tag{1.34}$$

```
> diff(f2(x), x);
```

$$\begin{cases} \mathit{undefined} & x = 0 \\ \mathit{undefined} & x = 1 \\ \mathit{undefined} & x = 2 \\ 0 & \mathit{otherwise} \end{cases} \tag{1.35}$$

```
> diff_f1 := Diff(f1(x), x);
```

$$\mathit{diff\_f1} := \frac{d}{dx}\left(\frac{1}{x^2+1}\right) \tag{1.36}$$

```
> eval(value(diff_f1), x = 1);
```

(1.37)

$$-\frac{1}{2} \quad (1.37)$$

```
> int(f1(x), x = -5..5);
```

$$2\arctan(5) \quad (1.38)$$

```
> evalf(int(f1(x), x = -5..5));
```

$$2.746801534 \quad (1.39)$$

```
> int(f1(x), x = -5..y);
```

$$\arctan(5) + \arctan(y) \quad (1.40)$$

```
> int(f1(x), x = -infinity..infinity); # nem srségfüggvény!
```

$$\pi \quad (1.41)$$

```
> int(f1(x), x); # határozatlan integrál
```

$$\arctan(x) \quad (1.42)$$

```
> Int(f2(x), x = -1..2.5);
```

$$\int_{-1}^{2.5} \begin{cases} 0 & x \leq 0 \\ \frac{1}{6} & x \leq 1 \\ \frac{1}{2} & x \leq 2 \\ 1 & \textit{otherwise} \end{cases} \,\mathrm{d}x \quad (1.43)$$

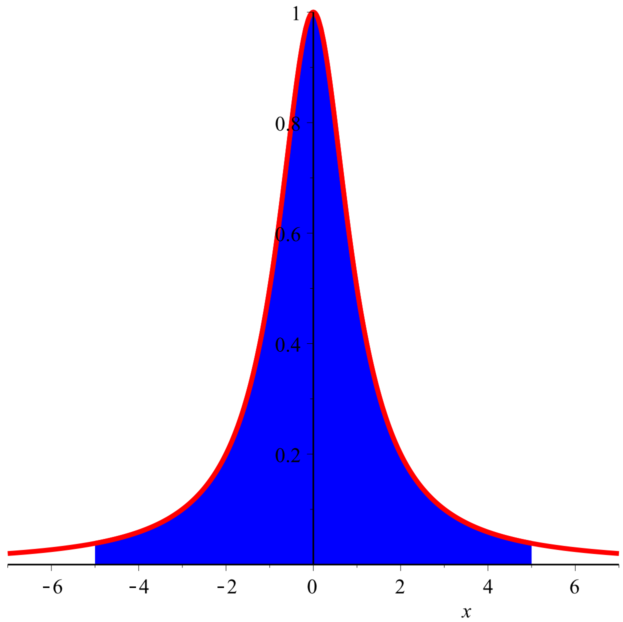
```
> value(%); # evalf(%) is mködik
```

$$1.166666667 \quad (1.44)$$

```
> P1 := plot(f1(x), x =-5..5, color = red, filled = [color =
  blue, transparency = 0.5], thickness = 3);
```

$$P1 := PLOT(\ldots) \quad (1.45)$$

```
> plots[display](P1, plot(f1(x), x = -7..7, color = red,
  thickness = 3));
```




```
> limit(f2(x), x = 2.5);
  f2(2.5); # folytonos
```

$$1.$$

$$1 \quad \textbf{(1.46)}$$

```
> limit(f2(x), x = 1);
```

$$\textit{undefined} \quad \textbf{(1.47)}$$

```
> limit(f2(x), x = 1, right);
```

$$\frac{1}{2} \quad \textbf{(1.48)}$$

```
> limit(f2(x), x = 1, left);
```

$$\frac{1}{6} \quad \textbf{(1.49)}$$

```
> # P(a <= X < b) = F(b) - F(a);
> # P(a < X <= b) = F(b+) - F(a+); # jobb oldali határérték
> Limit(f1(x), x = 1);
```

$$\lim_{x \to 1} \frac{1}{x^2 + 1} \quad \textbf{(1.50)}$$

```
> value(%); # evalf(%) is mködik
```

(1.51)

$$\frac{1}{2} \quad (1.51)$$

```
> limit(f1(x), x = infinity);
```

$$0 \quad (1.52)$$

```
> g := sin(x) + x; # nem függvény, hanem algebrai kifejezés! 
  'eval'-lal lehet értéket behelyettesíteni
```

$$g := \sin(x) + x \quad (1.53)$$

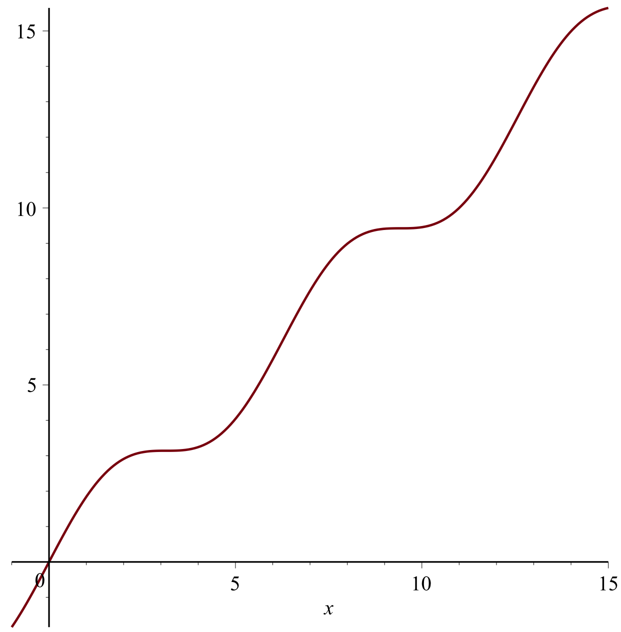
```
> g := unapply(sin(x) + x, x); # ez már függvény
```

$$g := x \rightarrow \sin(x) + x \quad (1.54)$$

```
> g := x -> sin(x) + x; # ez is
```

$$g := x \rightarrow \sin(x) + x \quad (1.55)$$

```
> plot(g(x), x = -1..15);
```



Nevezetes diszkrét eloszlások

Elméleti összefoglaló

Olykor részleteikben merben különböz feladatok is épülhetnek alapveten hasonló valószínségi modellre. Ha ezt felismerjük, és a közös rész kezelésére eszköztárat fejlesztünk ki,

sok hasonló feladatra nyerhetünk megoldási sablont, mellyel a feladattípus azonosítása –esetleg néhány paraméter illesztése– után könnyen eljuthatunk a megoldásig.
Hasonló valószínségi modellek hasonló *eloszlású* valószínségi változókkal írhatók le. Néhány gyakran elforduló nevezetes problémakörhöz *nevezetes eloszlások* tartoznak, melyek –a típus azonosítása után– megfelelő paraméterezéssel jól használhatók hasonló típusú feladatok megoldására. Ebben a feladatsorban a *nevezetes diszkrét eloszlások*kal foglalkozunk, azok közül is a következkkel: *diszkrét egyenletes*, *Bernoulli*, *binomiális*, *hipergeometrikus*, *Poisson* és *geometriai.*

Binomiális eloszlás

Legyen adott egy p valószínséggel sikeres (pozitív) kimenetel (és persze $q = 1 - p$ valószínséggel sikertelen (negatív) kimenetel) elemi kísérlet, melyet n-szer elvégzünk. Azt vizsgáljuk, hogy a kísérletsorozatban hány pozitív kísérletet találunk, és ezt egy Y valószínségi változóban rögzítjük.
Az Y valószínségi változó értékkészlete nyilván 0-tól n-ig terjed (egy sem sikerül, vagy mind sikerül, vagy ezek között bármilyen darabszám): $y = [0, 1, ..., n]$
Az egyes értékekhez tartozó valószínségek kiszámítása a következképp történik:
A $p_k = P(Y = k)$ érték azt a valószínséget jelenti, mellyel az n elem kísérletsorozatban pontosan k darab sikeres (és így pontosan $n - k$ darab sikertelen) kísérlet következik be. Az egymástól függetlenül elvégzett n elemi kísérletbl adott k db pozícióra (pl. konkrétan az els k-ra) elírva a sikert (a többire pedig a sikertelenséget), egy ilyen kimenetel kísérletsorozat valószínsége $p^k \cdot (1 - p)^{n-k}$, mivel a p valószínséggel bekövetkez pozitív eredménynek pontosan k-szor, az $1 - p$ valószínséggel bekövetkez negatív eredménynek pedig a maradék pozíciókban pontosan $(n - k)$-szor kell bekövetkeznie. Ez azonban csak egyetlen, elírt eredmény kísérletsorozatra vonatkozik (ahol pontosan elírt, hogy az n kísérlet közül épp melyik az a k darab, amelyik pozitív), miközben az adott k darabszámú sikeres kísérlet pozíciója nem kötött: az összes n kísérlet közül a k darab pozitív $\binom{n}{k}$ módon helyezkedhet el, tehát ennyiszer kell venni az elbbi valószínséget:

$$p_k = P(Y = k) = \binom{n}{k} p^k \cdot (1 - p)^{n-k}$$

Vegyük észre, hogy az eloszlásnak mindössze két (különböz feladatokra másképp beállítandó, de aztán az adott feladat során már konstans) paramétere van: az elemi kísérlet pozitív kimenetelének valószínsége (p), továbbá a kísérletsorozat elemszáma (n). Miután ezeket az adott feladatra megfelelen beállítottuk, Y bármely értékének valószínségét kiszámíthatjuk a formulával, ha a kívánt értéket k-ba helyettesítjük. Ebbl a szempontból n és p *paramétere*i, k pedig *változója* a fenti formulának.
Sok feladat visszavezethet binomiális sémára. Pl. a visszatevéses mintavétel, mint típusfeladat, maga is egy Bernoulli-kísérlet, hiszen a húzott elem visszahelyezésével a kísérlet minden ismétlésekor megegyezik a helyzet: ugyanannyi a pozitív ("selejtes"/megkülönböztetett) elem választásának valószínsége. Visszatevéses mintavétellel kiválasztott mintában a pozitív ("selejtes"/megkülönböztetett) elemek darabszáma binomiális eloszlást követ.
Adott típusú és paraméter eloszlás várható értéke és szórása is jól definiált, és csak az eloszlás paramétereitl függ.
A binomiális eloszlás várható értéke nagyon egyszer, szemléletes alakot ölt:

$$E(Y) = n \cdot p$$

Ez a formula szemléletesen azt fejezi ki, hogy a kísérletek közül átlagosan épp a valószínségnek megfelel arányban lesznek a pozitív kimenetelek.
A binomiális eloszlás varianciája és szórása:

$$Var(Y) = n \cdot p \cdot (1 - p)$$

$$\sigma(Y) = \sqrt{n \cdot p \cdot (1-p)}$$

Poisson-eloszlás

Tekintsünk egy olyan konstrukciót, ahol egy (végtelen hosszúságú) *egyenes mentén pontok szóródnak valamilyen átlagos egyenletes srséggel*. A gyakorlatban sok, ezzel modellezhet helyzet van; akár térben, akár idben (idtengelyen) szóródhatnak a megfigyelt, pontszernek tekinthet események (ill. azok valószínségi változóval számszersített értékei).
Ilyen modell szerint rendezdnek el pl. egy adott pillanatban egy útszakaszon az autók: ha légifelvételt készítünk az útszakaszról, akkor térben, de ha az amúgy mozgó jármveket egy megfigyelési ponton számláljuk, akkor idben is. Ezen analógia szerint értheten ugyanilyen modell szerint jönnek velünk szembe a járókelk az utcán, megfigyelt meteorok a légkörbe, vagy mszerrel érzékelt részecskék egy radioaktív forrásból.
A modell tipikusan arra az esetre használható, ha egy (térbeli vagy idbeli) szakaszon hosszabb távon *az átlagos pontsrség ismert* (elméleti úton meghatározható, vagy kísérletileg megmérhet). A pontok száma persze (adott hosszúságú szakaszokon újra és újra megszámolva) ezen átlag körül ingadozásokat mutat (a lehetséges darabszámokhoz tartozó megfelel valószínségekkel). Ezen eloszlás viszont nemhogy maga is pontosan meghatározott a konstrukció által, de ráadásul (érdekes és egyáltalán nem triviális módon) olyan szoros és egyértelm összefüggésben van a saját átlagával, hogy a teljes szerkezete maradéktalanul jellemezhet a várható értékkel, mint egyetlen paraméterrel (λ) – a megfigyelt szakaszon mért átlagos darabszámon kívül semmilyen más információra nincs szükség az elfordulási darabszámok valószínségének leírásához.
A konstrukcióban tipikusan feltehet kérdések:
1. Egy adott szakaszon hány pontra milyen valószínséggel lehet számítani? (Adott hosszúságú szakaszon mérhet darabszám valószínségi eloszlása.)
2. Az egyes pontok között mekkora távolságra milyen valószínséggel lehet számítani? (Szomszédos pontok közti távolság valószínségi eloszlása.)
Az els kérdésre a *Poisson-eloszlás*, a másodikra a késbb sorra kerül *exponenciális eloszlás* ad választ. A kett jelentsen eltér már abban is, hogy elbbi diszkrét, utóbbi folytonos, mégis összefüggnek a közös modell révén.
A Maple képes maga is létrehozni és kezelni Poisson-eloszlású valószínségi változót:

```
> restart;
> with(Statistics):
> Y := RandomVariable(Poisson(4));
```

$$Y := \_R \qquad \textbf{(2.1.2.1)}$$

A λ paraméter Poisson-eloszlás várható értéke és varianciája (szórásnégyzete) is maga a paraméter: $\mu = \sigma^2 = \lambda$:

```
> Mean(Y);
```

$$4 \qquad \textbf{(2.1.2.2)}$$

```
> Variance(Y);
```

$$4 \qquad \textbf{(2.1.2.3)}$$

```
> StandardDeviation(Y);
```

$$2 \qquad \textbf{(2.1.2.4)}$$

Kidolgozott feladatok

1. Feladat (Poisson-eloszlás a binomiális eloszlás határértékeként)

Szórjunk le n pontot véletlenszeren egy $l = n$ hosszúságú szakaszra! Válasszunk ki egy $d = 1$ hosszúságú részszakaszt és jelölje Y az erre es pontok számát! Viszgáljuk Y

eloszlását *n* különböz értékei esetén!
a) Adjuk meg *Y* eloszlásának típusát, paramétereit és várható értékét!
b) Határozzuk meg és ábrázoljuk pálcikadiagramon *Y* eloszlását $n = 10$ esetén! Számítsuk ki a várható értéket és a szórást!
c) Határozzuk meg és ábrázoljuk pálcikadiagramon *Y* eloszlását $n = 100$ esetén! Számítsuk ki a várható értéket és a szórást!
d) Közelítsük *Y* eloszlását Poisson-eloszlással, adjuk meg a λ paraméter értékét! Ábrázoljuk a valószínségeket a tartomány alkalmas megválasztása mellett pálcikadiagramon és hasonlítsuk ezt össze a c) feladatrészben kapott diagrammal!

Megoldás

```
> restart;
```

a) Adjuk meg *Y* eloszlásának típusát, paramétereit és várható értékét!
Elvileg a teljesen belül bárhol kiszemelhetjük a vizsgált 1 hosszú szakaszt, pl. képzeljük valahova középre.
Könnyen látható, hogy a kísérletet elvégezve a kiszemelt szakaszon belüli pontok száma binomiális eloszlású: egy pont ledobása egy Bernoulli-kísérlet, mely –geometriai valószínűséggel számolva– $p = \frac{d}{l} = \frac{1}{n}$ valószínséggel lesz sikeres (a pont a kiszemelt szakaszra esik), és *Y* éppen *n* db ilyen, függetlenül elvégzett kísérlet közül a sikeresek száma. Az eloszlás paraméterei *n* és $p = \frac{1}{n}$, vagyis $Y{\sim}Binomial\left(n, \frac{1}{n}\right)$.

Y várható értéke a binomiális eloszlásra tanult képlet alapján $E(Y) = n \cdot p = \frac{n}{n} = 1$, ami jelen esetben nem függ *n*-t1.

```
> EY := 1;
```

$$EY := 1 \qquad \textbf{(2.2.1.1.1)}$$

b) Határozzuk meg és ábrázoljuk pálcikadiagramon *Y* eloszlását $n = 10$ esetén! Számítsuk ki a várható értéket és a szórást!

```
> n := 10; l := n; d := 1;
```

$$n := 10$$
$$l := 10$$
$$d := 1 \qquad \textbf{(2.2.1.1.2)}$$

```
> p := d/l;
```

$$p := \frac{1}{10} \qquad \textbf{(2.2.1.1.3)}$$

Y lehetséges értékei a 0-10 számok (szélsséges esetben elfordulhat, hogy egy pont sem esik a kiválasztott szakaszra, és az is, hogy mind a 10 pont ráesik).

```
> y10 := [seq(i, i = 0..n)];
```

$$y10 := [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] \qquad \textbf{(2.2.1.1.4)}$$

Az egyes értékekhez tartozó valószínségek binomiális eloszlás szerint alakulnak:

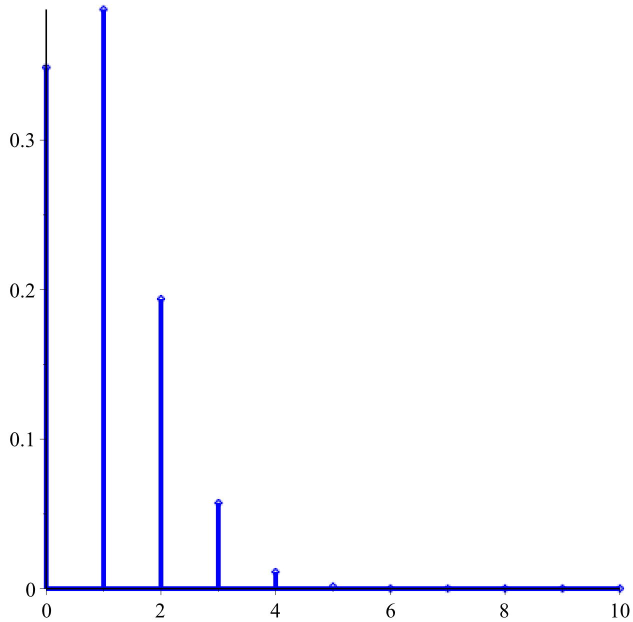
```
> py10 := [seq(binomial(n, k)*p^k*(1-p)^(n-k), k = 0..n)]:
  evalf(%);
```

$$[0.3486784401, 0.3874204890, 0.1937102445, 0.05739562800, \qquad \textbf{(2.2.1.1.5)}$$
$$0.01116026100, 0.001488034800, 0.0001377810000,$$
$$0.000008748000000, 3.645000000\ 10^{-7}, 9.000000000\ 10^{-9},$$

$1.000000000\ 10^{-10}]$

Ábrázoljuk az eloszlást pálcikadiagramon!

```
> with(DynamicSystems):
> P1 := DiscretePlot(y10, py10, style = stem, color =
  blue, thickness = 3): P1;
```



A várható értéket és a szórást számolhatjuk definíció szerint vagy a Statistics csomag beépített eljárásaival. A várható értékre 1-et kell kapnunk az a) feladatrész alapján.

```
> EY10 := sum(y10[i + 1]*py10[i + 1], i = 0..n); evalf(%);
  # az indexek el vannak tolva 1-gyel, mert a lista 1-tl
  indexeldik
```

$$EY10 := 1$$

$$1. \qquad \textbf{(2.2.1.1.6)}$$

A szórás:

```
> sigmaY10 := sqrt(sum((y10[i + 1] - EY10)^2*py10[i + 1],
  i = 0..n)); evalf(%);
```

$$sigmaY10 := \frac{3}{10}\sqrt{10}$$

$$0.9486832980 \qquad \textbf{(2.2.1.1.7)}$$

A Statistics csomag eszközeivel:

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

```
> Y := RandomVariable(Binomial(n, p));
```
$$Y := \_R \tag{2.2.1.1.8}$$

```
> ExpectedValue(Y); # Mean(Y) is mködik
```
$$1 \tag{2.2.1.1.9}$$

```
> StandardDeviation(Y);
```
$$\frac{3}{10}\sqrt{10} \tag{2.2.1.1.10}$$

Az eloszlást számolhatjuk így is:

```
> evalf([seq(ProbabilityFunction(Y, k), k = 0..n)]); # =
  py10
```
$[0.3486784401, 0.3874204890, 0.1937102445, 0.05739562800, 0.01116026100, 0.001488034800, 0.0001377810000, 0.000008748000000, 3.645000000\ 10^{-7}, 9.000000000\ 10^{-9}, 1.000000000\ 10^{-10}]$ **(2.2.1.1.11)**

c) Határozzuk meg és ábrázoljuk pálcikadiagramon *Y* eloszlását $n = 100$ esetén! Számítsuk ki a várható értéket és a szórást!

```
> n := 100; l := n; d := 1;
```
$$n := 100$$
$$l := 100$$
$$d := 1 \tag{2.2.1.1.12}$$

```
> p := d/l;
```
$$p := \frac{1}{100} \tag{2.2.1.1.13}$$

Y értékei most a 0-100 számok közül kerülnek ki (szélsséges esetben elfordulhat, hogy egy pont sem esik a kiválasztott szakaszra, és az is, hogy az összes pont ráesik).

```
> y100 := [seq(i, i = 0..n)];
```
$y100 := [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]$ **(2.2.1.1.14)**

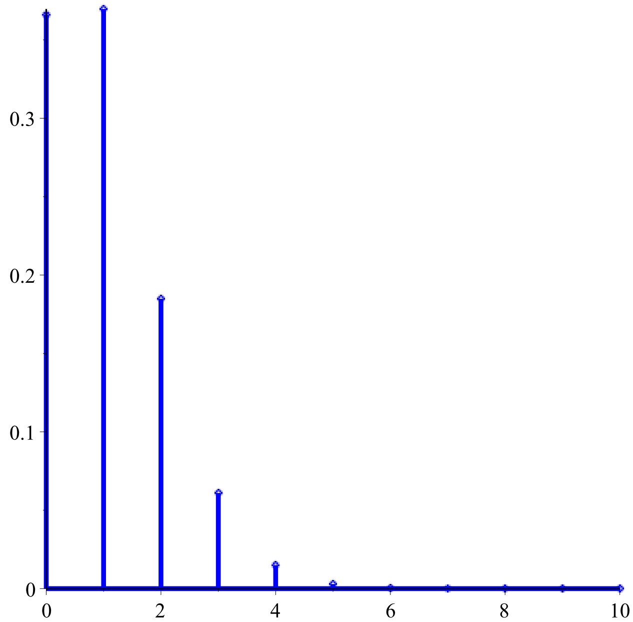
Az egyes értékekhez tartozó valószínségek binomiális eloszlás szerint alakulnak:

```
> py100 := [seq(binomial(n, k)*p^k*(1-p)^(n-k), k = 0..n)]
  : evalf(%);
```
$[0.3660323413, 0.3697296376, 0.1848648188, 0.06099916581, 0.01494171486, 0.002897787124, 0.0004634508026, 0.00006286345663, 0.000007381693771, 7.621950920\ 10^{-7}, 7.006035694\ 10^{-8}, 5.790112144\ 10^{-9}, 4.337710276\ 10^{-10}, 2.965955744\ 10^{-11}, 1.861747112\ 10^{-12}, 1.078183513\ 10^{-13}, 5.785706982\ 10^{-15}, 2.887696889\ 10^{-16}, 1.344999112\ 10^{-17}, 5.863366678\ 10^{-19}, 2.398650004\ 10^{-20}, 9.230014447\ 10^{-22}, 3.347893211\ 10^{-23}, 1.146840889\ 10^{-24}, 3.716613992\ 10^{-26},$ **(2.2.1.1.15)**

$1.141263287\ 10^{-27}$, $3.325359227\ 10^{-29}$, $9.206007586\ 10^{-31}$,
$2.424381507\ 10^{-32}$, $6.079953623\ 10^{-34}$, $1.453456927\ 10^{-35}$,
$3.315151022\ 10^{-37}$, $7.220499385\ 10^{-39}$, $1.502889373\ 10^{-40}$,
$2.991491028\ 10^{-42}$, $5.698078148\ 10^{-44}$, $1.039211783\ 10^{-45}$,
$1.815712644\ 10^{-47}$, $3.040667107\ 10^{-49}$, $4.882708123\ 10^{-51}$,
$7.521343321\ 10^{-53}$, $1.111802413\ 10^{-54}$, $1.577593611\ 10^{-56}$,
$2.149411074\ 10^{-58}$, $2.812590249\ 10^{-60}$, $3.535466979\ 10^{-62}$,
$4.269887656\ 10^{-64}$, $4.955382193\ 10^{-66}$, $5.526836200\ 10^{-68}$,
$5.924458511\ 10^{-70}$, $6.103987557\ 10^{-72}$, $6.044749017\ 10^{-74}$,
$5.753548986\ 10^{-76}$, $5.263395299\ 10^{-78}$, $4.627377087\ 10^{-80}$,
$3.909262553\ 10^{-82}$, $3.173102721\ 10^{-84}$, $2.474154169\ 10^{-86}$,
$1.852814860\ 10^{-88}$, $1.332275708\ 10^{-90}$, $9.195842430\ 10^{-93}$,
$6.090970313\ 10^{-95}$, $3.870117990\ 10^{-97}$, $2.357936245\ 10^{-99}$,
$1.376951406\ 10^{-101}$, $7.703224648\ 10^{-104}$, $4.126306438\ 10^{-106}$,
$2.115097526\ 10^{-108}$, $1.036812513\ 10^{-110}$, $4.856975612\ 10^{-113}$,
$2.172673073\ 10^{-115}$, $9.273039152\ 10^{-118}$, $3.772701114\ 10^{-120}$,
$1.461680243\ 10^{-122}$, $5.387027924\ 10^{-125}$, $1.886366681\ 10^{-127}$,
$6.267831873\ 10^{-130}$, $1.973343368\ 10^{-132}$, $5.877609100\ 10^{-135}$,
$1.653335893\ 10^{-137}$, $4.383845171\ 10^{-140}$, $1.093364552\ 10^{-142}$,
$2.558995625\ 10^{-145}$, $5.605685926\ 10^{-148}$, $1.145943491\ 10^{-150}$,
$2.178858688\ 10^{-153}$, $3.838722143\ 10^{-156}$, $6.239650529\ 10^{-159}$,
$9.310773286\ 10^{-162}$, $1.268065820\ 10^{-164}$, $1.565513359\ 10^{-167}$,
$1.737721566\ 10^{-170}$, $1.717116172\ 10^{-173}$, $1.492009273\ 10^{-176}$,
$1.122293672\ 10^{-179}$, $7.159768240\ 10^{-183}$, $3.766713089\ 10^{-186}$,
$1.568973483\ 10^{-189}$, $4.851495000\ 10^{-193}$, $9.900000000\ 10^{-197}$,
$1.000000000\ 10^{-200}$]

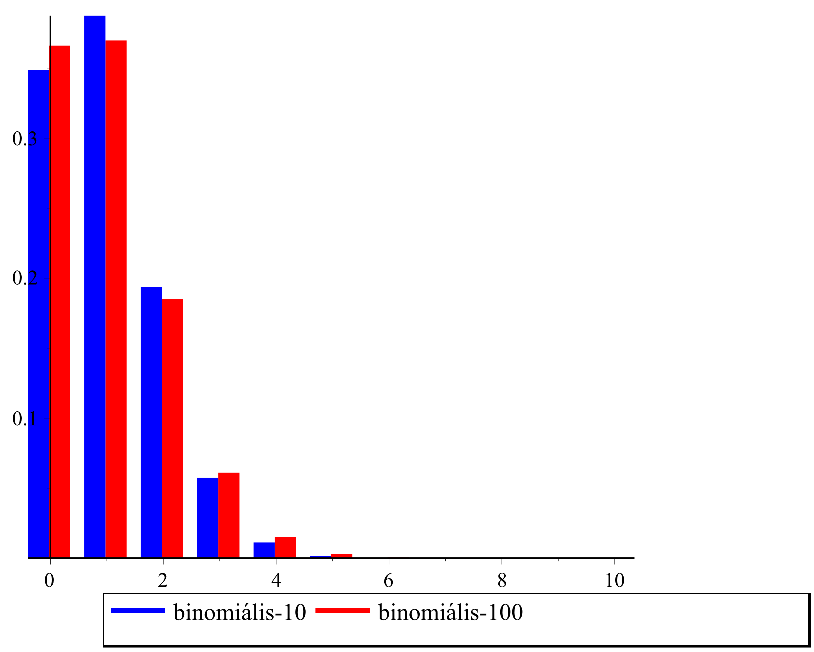
Noha látszik, hogy gyakorlatilag 5-nél több pont szinte soha nem srsödik 1 méteren belül, elvi fontosságú felismerés, hogy ez nem a kísérleti konstrukció korlátaiból következik (hiszen a gyakorlati korlát jóval efölött, 100-nál van), hanem a kisérlet alapvet természetébl fakad.
Ábrázoljuk az eloszlás érdemi részét pálcikadiagramon, pl. szorítkozzunk csak a 0-10 értékekre!

```
> P2 := DiscretePlot(y100[1..11], py100[1..11], style =
  stem, color = blue, thickness = 3): P2; # indexek 1-gyel
  eltolva!
```



Ha ezt összehasonlítjuk a b) feladatrészben, 10 hosszúságú szakaszon kapott diagrammal, látszik, hogy alapveten hasonlóak, bár de vannak apróbb különbségek (csak az els két pálcika magasságában van látható eltérés). Ez nem véletlen, mint azt késbb látni fogjuk.

```
> ColumnGraph([py10, py100[1..11]], color = [blue, red],
  offset = -0.4, legend = ["binomiális-10",
  "binomiális-100"]); # ha több adatsort ábrázolunk,
  mindig készítsünk "legend"-et!
```



A várható értéket és a szórást számolhatjuk definíció alapján vagy a Statistics csomag beépített eljárásaival. A várható értékre 1-et kell kapnunk az a) feladatrész szerint.

```
> EY100 := sum(y100[i + 1]*py100[i + 1], i = 0..n); evalf
  (%); # indexek 1-gyel eltolva
```

$$EY100 := 1$$

$$1. \quad \textbf{(2.2.1.1.16)}$$

A szórás:

```
> sigmaY100 := sqrt(sum((y100[i + 1] - EY100)^2*py100[i +
  1], i = 0..n)); evalf(%);
```

$$sigmaY100 := \frac{3}{10}\sqrt{11}$$

$$0.9949874370 \quad \textbf{(2.2.1.1.17)}$$

A Statistics csomag eszközeivel:

```
> Y := RandomVariable(Binomial(n, p));
```

$$Y := \_R0 \quad \textbf{(2.2.1.1.18)}$$

```
> ExpectedValue(Y); # Mean(Y) is mködik
```

$$1 \quad \textbf{(2.2.1.1.19)}$$

```
> StandardDeviation(Y);
```

(2.2.1.1.20)

$$\frac{3}{10}\sqrt{11} \qquad (2.2.1.1.20)$$

d) Közelítsük *Y* eloszlását Poisson-eloszlással, adjuk meg a λ paraméter értékét! Ábrázoljuk a valószínségeket a tartomány alkalmas megválasztása mellett pálcikadiagramon és hasonlítsuk ezt össze a c) feladatrészben kapott diagrammal!

A véges intervallumon vett gyakorlati modellt a végtelen számegyenesen vett elméleti modellel szokás közelíteni.

Növelve a leszórt pontok számát és ezzel együtt a szakasz hosszát (pl. 1000 méteren 1000 pont, 1000000 méteren 1000000 pont, stb.), az 1 egység hosszú szakszra es pontok *Y* számának eloszlása stabilizálódni látszik egy határeloszlás körül, ahol az egész végtelen számegyenesre szórunk le végtelen sok pontot valamilyen átlagos sűrűséggel – jelen esetben úgy, hogy bármely 1 hosszúságú szakaszra várhatóan 1 pont essen.

Azt is gondolhatnánk, hogy a véges 10 egység hosszon egyenletes eloszlással elszórt 10 pont mintázata tökéletesen megegyezik azzal a mintázattal, mely egy végtelen hosszú egyenes mentén bármely 10 hosszú szakaszon keletkezne méterenként 1 darabos átlagos pontsrség mellett. Bár való igaz, hogy közelíti, világos, hogy teljesen pontosan nem írhatja le, hiszen az $n = 10$ esetben kizárt, hogy a megfigyelt 1 méteres szakaszra 10-nél több pont essen, miközben a végtelen hosszú egyenesre szórt végtelen sok pont esetén semmi nem tiltja, hogy egy kiszemelt 1 méteres szakaszon véletlenszer lokális srsödés folytán 10-nél több (st, elvileg bármennyi) pont jelenjen meg.

Az elméletileg is teljesen pontos leíráshoz nyilván olyan eloszlás kell, mely elvileg egyáltalán nem korlátozza a megfigyelt szakaszon a pontok számát, de persze az átlagtól való nagyobb eltérések mellé megfelelen kis valószínséget társít. Ez az elméletileg tökéletes leírást adó eloszlás a *Poisson-eloszlás*.

Ennek formulája a következ:

```
> poisson_eloszlas := (lambda, k) -> lambda^k/k!*exp(-lambda);
```

$$poisson\_eloszlas := (\lambda, k) \rightarrow \frac{\lambda^k e^{-\lambda}}{k!} \qquad (2.2.1.1.21)$$

A formula konstans paramétere az a λ érték, mely a megfigyelt szakaszon megjelen pontok átlagos száma. Ha *k* helyére beírunk egy tetszleges nemnegatív számot (merthogy zérus lehet az elfordulási darabszám, de negatív nem), akkor a Poisson-féle formula megadja, hogy mekkora valószínséggel találunk éppen *k* darab pontot a kiszemelt intervallumon. Vegyük észre, hogy a formula tetszlegesen nagy *k* számra pozitív valószínséget ad, tehát jól illeszkedik ahhoz a helyzethez, hogy a modellben leszórt végtelen sok pont mellett elvileg nem zárható ki bármennyinek a felhalmozódása egy adott intervallumon (persze megfelelen kicsiny valószínséggel: a legnagyobb valószínségeket a λ közelében lév *k* értékekre kapjuk, ennél nagyobb értékek elfordulási valószínsége rohamosan csökken.)

```
> lambda := EY; # pontok átlagsrsége (1 hosszú szakaszra átlag hány pont esik)
  korlat := 10; # releváns értékek k=10-ig
```

$$\lambda := 1$$

$$korlat := 10 \qquad (2.2.1.1.22)$$

```
> y := [seq(i, i = 0..korlat)];
```

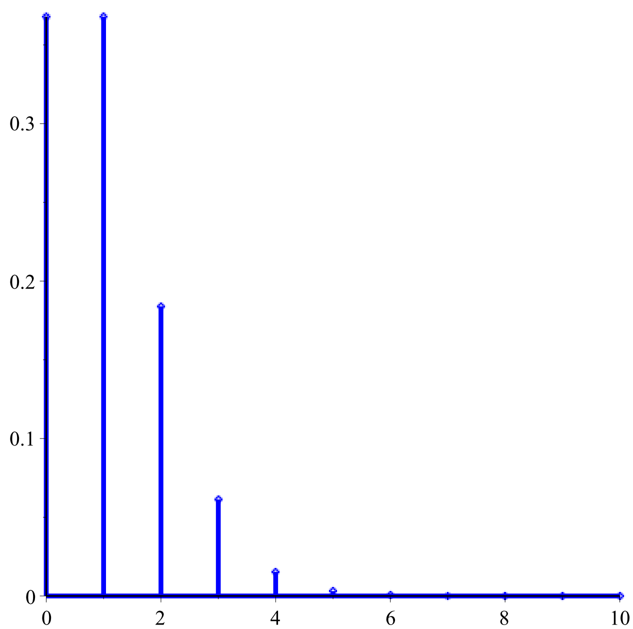
$$y := [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] \qquad (2.2.1.1.23)$$

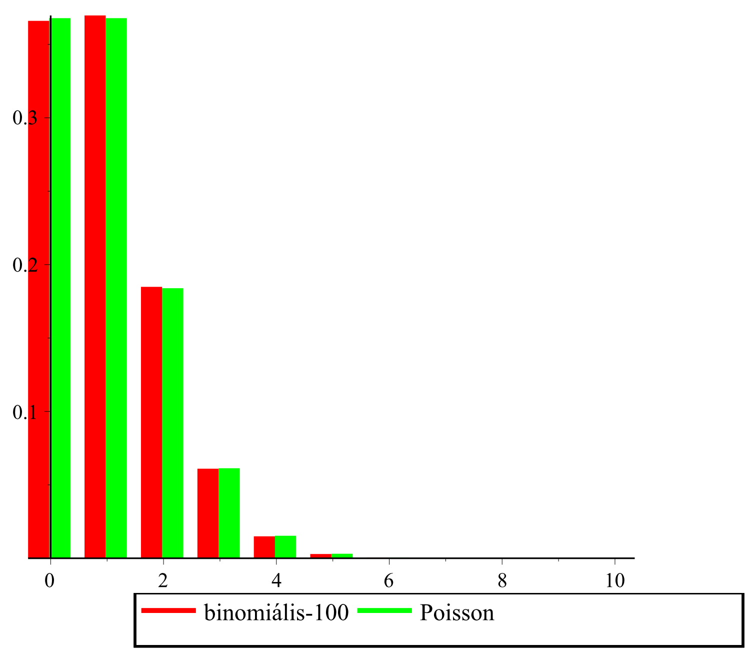
```
> py := [seq(poisson_eloszlas(lambda, i), i = 0..korlat)];
  evalf(%);
```

$$py := \left[e^{-1}, e^{-1}, \frac{1}{2}\, e^{-1}, \frac{1}{6}\, e^{-1}, \frac{1}{24}\, e^{-1}, \frac{1}{120}\, e^{-1}, \frac{1}{720}\, e^{-1}, \frac{1}{5040}\, e^{-1}, \frac{1}{40320}\, e^{-1}, \frac{1}{362880}\, e^{-1}, \frac{1}{3628800}\, e^{-1}\right]$$

$$[0.3678794412, 0.3678794412, 0.1839397206, 0.06131324021, 0.01532831005, 0.003065662010, 0.0005109436684, 0.00007299195261, 0.000009123994077, 0.000001013777120, 1.013777120\ 10^{-7}] \tag{2.2.1.1.24}$$

```
> DiscretePlot(y, evalf(py), style = stem, color = blue,
  thickness = 3);
  ColumnGraph([py100[1..11], py], color = [red, green],
  offset = -0.4, legend = ["binomiális-100", "Poisson"]);
```





A Poisson-eloszlás (piros) és a 100-as binomiális eloszlás megfelel értékei (zöld) meggyz egyezést mutatnak.
Figyelmet érdemel, hogy a várható értékre a modellbl visszaszámolva kijön az az 1 érték, amire az egész konstrukciót eredetileg is építettük:

```
> EPoisson := sum(k*poisson_eloszlas(lambda, k), k = 0..
  infinity); # végtelenig történik a szummázás!
```

$$EPoisson := 1 \qquad \textbf{(2.2.1.1.25)}$$

A repültársaságoknál bevett gyakorlat, hogy statisztikai adatok és valószínségi meggondolások alapján a járatokra olykor a gép befogadóképességénél több jegyet adnak el. Statisztikákkal mérhet tapasztalat ugyanis, hogy a jegyeket megvásárló utasok egy része végül nem száll fel, mert valami közbejön, és lemondja/elhalasztja az utat, esetleg egyszeren lekési a járatot, a kihasználatlan helyek pedig a cégnek bevétel-kiesést jelentenek. (Lehet, hogy részben vagy egészben vissza is térítenek a kimaradó utasnak, de még ha nem, akkor is további bevételt jelent, ha a megüresed helyekre újabb fizet utasokat ültetnek.)
Persze ha több jegyet adnak el, azzal azt kockáztatják, hogy esetleg nem minden fizet utas fér fel, és ha ez a jelenség túl gyakorivá válik, akkor az utasok bizalma megrendül, elfordulnak a cégtl, kártérítést követelnek stb. A járat befogadóképességén felül eladható jegyek mennyiségét úgy kell meghatározni, hogy amellett, hogy a gép minél jobb kihasználtsággal repüljön, azért nagy valószínséggel felférjen mindenki. Garancia erre persze csak akkor van, ha nem adunk el

több jegyet, de ekkor nagy valószínséggel maradnak biztos bevétel-kiesést jelent üres helyek, ami biztos veszteség, így amíg nem válik túl naggyá a létszám-túllépés esélye, addig érdemes lehet kockáztatni, túltervezni. (Persze néha-néha lesz, aki nem fér fel, de lehet, hogy ez annyira ritka, hogy az ilyen esetben kifizetend kártérítés ill. az ebbl adódó presztízs-veszteség mértékét a plusz jegyek eladásából származó bevétel még így is bven kompenzálja.)
A következkben valószínségi modellvizsgálatot végzünk arra, hogy mennyi plusz jegy eladása milyen kockázattal jár. Tekintsük a következ helyzetet:

2. Feladat (Repüljegyek eladása / 1.)

Tegyük fel, hogy egy járat befogadóképessége 120 f. Tapasztalatok szerint a jeggyel rendelkez utasok átlagosan 10%-a nem száll fel a gépre.
a) Tegyük fel, hogy 125 jegyet adnak el a gépre. Adjuk meg a beszálló utasok számának valószínségi eloszlását!
b) Adjuk meg az eloszlás várható értékét és szórását!
c) Ábrázoljuk az eloszlást pálcikadiagramon (a tartomány megfelel megválasztása mellett)!
d) Átlagosan hány utas nem jelenik meg az indulásnál?
e) Mekkora a valószínsége, hogy 125 eladott jegy esetén az összes (indulásig megjelen) utas felfér?
f) Mennyi a valószínsége, hogy marad üres ülés a járaton?

Megoldás

```
> restart;
```

Vegyük észre, hogy egy-egy, jeggyel rendelkez utas viselkedése egy-egy kétesélyes Bernoulli-kísérlet, jól meghatározható valószínségekkel: a megjelenés valószínsége $p = 90\ \%$, a távolmaradásé $1 - p = 10\ \%$. Ezt úgy is elképzelhetjük, hogy amikor várjuk az utasokat a beszállásnál, minden jeggyel rendelkez utasra elvégezzük a kísérletet. A gépnél végül megjelen utasok száma így a 125 elem Bernoulli-kísérletsorozatban a pozitív kimenetel kísérletek száma. Ez a szám mint valószínségi változó (Y) –a konstrukcióból következően– olyan binomiális eloszlású, melynek paraméterei: $n = 125$ és $p = 0.9$.
a) Tegyük fel, hogy 125 jegyet adnak el a gépre. Adjuk meg a beszálló utasok számának valószínségi eloszlását!

```
> n := 125; p := 0.9;
```

$$n := 125$$
$$p := 0.9 \qquad (2.2.2.1.1)$$

A binomiális eloszlást leíró formula:

```
> binomialis := k -> binomial(n, k)*p^k*(1-p)^(n-k);
```

$$binomialis := k \rightarrow \mathrm{binomial}(n, k)\, p^k\, (1-p)^{n-k} \qquad (2.2.2.1.2)$$

A valószínségi változó értékkészlete:

```
> y := [seq(i, i = 0..125)];
```

$$y := [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125] \qquad (2.2.2.1.3)$$

Az eloszlás (a fentebbi formulát alkalmazva az értékkészlet összes elemére):

```
> py := [seq(binomialis(k), k = 0..n)];
```

$py := [1.0\ 10^{-125}, 1.125\ 10^{-122}, 6.27750\ 10^{-120}, 2.31639750\ 10^{-117},$ **(2.2.2.1.4)**

$6.358511138\ 10^{-115}, 1.384883726\ 10^{-112}, 2.492790706\ 10^{-110},$
$3.813969781\ 10^{-108}, 5.063044883\ 10^{-106}, 5.923762513\ 10^{-104},$
$6.184408065\ 10^{-102}, 5.818965770\ 10^{-100}, 4.975215733\ 10^{-98},$
$3.892149538\ 10^{-96}, 2.802347670\ 10^{-94}, 1.866363547\ 10^{-92},$
$1.154812445\ 10^{-90}, 6.663947107\ 10^{-89}, 3.598531438\ 10^{-87},$
$1.823887250\ 10^{-85}, 8.699942183\ 10^{-84}, 3.914973980\ 10^{-82},$
$1.665643475\ 10^{-80}, 6.713267401\ 10^{-79}, 2.567824781\ 10^{-77},$
$9.336610900\ 10^{-76}, 3.231903774\ 10^{-74}, 1.066528246\ 10^{-72},$
$3.359563973\ 10^{-71}, 1.011344603\ 10^{-69}, 2.912672457\ 10^{-68},$
$8.033338551\ 10^{-67}, 2.123813879\ 10^{-65}, 5.386764292\ 10^{-64},$
$1.311835539\ 10^{-62}, 3.069695164\ 10^{-61}, 6.906814115\ 10^{-60},$
$1.495231921\ 10^{-58}, 3.116378108\ 10^{-57}, 6.256728357\ 10^{-56},$
$1.210676937\ 10^{-54}, 2.258945992\ 10^{-53}, 4.066102785\ 10^{-52},$
$7.063671581\ 10^{-51}, 1.184770370\ 10^{-49}, 1.919328000\ 10^{-48},$
$3.004165565\ 10^{-47}, 4.544599397\ 10^{-46}, 6.646476616\ 10^{-45},$
$9.400016933\ 10^{-44}, 1.285922316\ 10^{-42}, 1.701956007\ 10^{-41},$
$2.179812886\ 10^{-40}, 2.702145407\ 10^{-39}, 3.242574489\ 10^{-38},$
$3.767281997\ 10^{-37}, 4.238192247\ 10^{-36}, 4.617398921\ 10^{-35},$
$4.872151964\ 10^{-34}, 4.979504465\ 10^{-33}, 4.929709422\ 10^{-32},$
$4.727672150\ 10^{-31}, 4.392159933\ 10^{-30}, 3.952943941\ 10^{-29},$
$3.446472998\ 10^{-28}, 2.910944117\ 10^{-27}, 2.381681549\ 10^{-26},$
$1.887571497\ 10^{-25}, 1.448988708\ 10^{-24}, 1.077291605\ 10^{-23},$
$7.756499554\ 10^{-23}, 5.407700392\ 10^{-22}, 3.650197765\ 10^{-21},$
$2.385129224\ 10^{-20}, 1.508433077\ 10^{-19}, 9.231610431\ 10^{-19},$
$5.466085124\ 10^{-18}, 3.130576025\ 10^{-17}, 1.733857491\ 10^{-16},$
$9.283819223\ 10^{-16}, 4.804376448\ 10^{-15}, 2.402188224\ 10^{-14},$
$1.160081142\ 10^{-13}, 5.409053038\ 10^{-13}, 2.434073867\ 10^{-12},$
$1.056674420\ 10^{-11}, 4.423288268\ 10^{-11}, 1.784568025\ 10^{-10},$
$6.935480283\ 10^{-10}, 2.594960600\ 10^{-9}, 9.341858163\ 10^{-9},$
$3.233720132\ 10^{-8}, 1.075563435\ 10^{-7}, 3.434863874\ 10^{-7},$
$0.000001052383825, 0.000003090685129, 0.000008692551926,$
$0.00002338923765, 0.00006014375398, 0.0001476255779,$
$0.0003454438523, 0.0007695531364, 0.001629641936, 0.003275105637,$
$0.006235297272, 0.01122353509, 0.01905883318, 0.03045850910,$

0.04568776362, 0.06413053063, 0.08395269461, 0.1021046286, 0.1148677071, 0.1189338207, 0.1126741459, 0.09699774300, 0.07525686955, 0.05210090970, 0.03179038558, 0.01683020413, 0.007573591859, 0.002816625071, 0.0008311352671, 0.0001824443269, 0.00002648385390, 0.000001906837481]

Ellenrizhetjük is a valószínségek összegét, hogy valóban eloszlásról van-e szó:

```
> sum(py[i], i = 1..nops(py)); #kerekítési hibák miatt picivel eltérhet 1-tl
```

$$0.9999999999 \quad (2.2.2.1.5)$$

b) Adjuk meg az eloszlás várható értékét és szórását!

A várható érték:

```
> EY := sum(y[i]*py[i], i = 1..nops(y));
```

$$EY := 112.5000001 \quad (2.2.2.1.6)$$

Vegyük észre, hogy az általános számítás eredménye megegyezik a binomiális eloszlás formula szerinti várható értékével.

```
> EY := n*p;
```

$$EY := 112.5 \quad (2.2.2.1.7)$$

Fejben számolva is könnyen adódik, hogy a 125 utasból átlagosan 12.5 (10 %) hiányzik, ezért 112.5 a gépnél megjelenk átlagos száma.

Számítsuk ki a szórást is!

```
> sigmaY := sqrt(sum((y[i]-EY)^2*py[i], i = 1..nops(y)));
```

$$sigmaY := 3.354101965 \quad (2.2.2.1.8)$$

Vegyük észre, hogy az általános számítás eredménye megegyezik a binomiális eloszlás formula szerinti szórásával:

```
> sigmaY := sqrt(n*p*(1-p));
```

$$sigmaY := 3.354101966 \quad (2.2.2.1.9)$$

Ugyanezt kiszámíthatjuk a Maple magasabb szint eljárásaival.

Hozzunk létre egy megfelelen paraméterezett binomiális eloszlású valószínségi változót:

```
> with(Statistics):
> Y := RandomVariable(Binomial(n, p));
```

$$Y := \_R \quad (2.2.2.1.10)$$

Beépített Maple függvénnyel a változó bármely értékére lekérdezhet a valószínség. Pl. $k = 0$ értéknél:

```
> ProbabilityFunction(Y, 0);
```

$$1.0\ 10^{-125} \quad (2.2.2.1.11)$$

Az értékkészlet a fentivel megegyez, de a valószínségeket ezzel a függvénnyel is generálhatjuk:

```
> py := [seq(ProbabilityFunction(Y, k), k = 0..125)];
```

$py := [1.0\ 10^{-125}, 1.125\ 10^{-122}, 6.27750\ 10^{-120}, 2.31639750\ 10^{-117}, 6.358511138\ 10^{-115}, 1.384883726\ 10^{-112}, 2.492790706\ 10^{-110}, 3.813969781\ 10^{-108}, 5.063044883\ 10^{-106}, 5.923762513\ 10^{-104}, 6.184408065\ 10^{-102}, 5.818965770\ 10^{-100}, 4.975215733\ 10^{-98}, 3.892149538\ 10^{-96}, 2.802347670\ 10^{-94}, 1.866363547\ 10^{-92}, 1.154812445\ 10^{-90}, 6.663947107\ 10^{-89}, 3.598531438\ 10^{-87},$ (2.2.2.1.12)

$1.823887250\ 10^{-85}$, $8.699942183\ 10^{-84}$, $3.914973980\ 10^{-82}$,
$1.665643475\ 10^{-80}$, $6.713267401\ 10^{-79}$, $2.567824781\ 10^{-77}$,
$9.336610900\ 10^{-76}$, $3.231903774\ 10^{-74}$, $1.066528246\ 10^{-72}$,
$3.359563973\ 10^{-71}$, $1.011344603\ 10^{-69}$, $2.912672457\ 10^{-68}$,
$8.033338551\ 10^{-67}$, $2.123813879\ 10^{-65}$, $5.386764292\ 10^{-64}$,
$1.311835539\ 10^{-62}$, $3.069695164\ 10^{-61}$, $6.906814115\ 10^{-60}$,
$1.495231921\ 10^{-58}$, $3.116378108\ 10^{-57}$, $6.256728357\ 10^{-56}$,
$1.210676937\ 10^{-54}$, $2.258945992\ 10^{-53}$, $4.066102785\ 10^{-52}$,
$7.063671581\ 10^{-51}$, $1.184770370\ 10^{-49}$, $1.919328000\ 10^{-48}$,
$3.004165565\ 10^{-47}$, $4.544599397\ 10^{-46}$, $6.646476616\ 10^{-45}$,
$9.400016933\ 10^{-44}$, $1.285922316\ 10^{-42}$, $1.701956007\ 10^{-41}$,
$2.179812886\ 10^{-40}$, $2.702145407\ 10^{-39}$, $3.242574489\ 10^{-38}$,
$3.767281997\ 10^{-37}$, $4.238192247\ 10^{-36}$, $4.617398921\ 10^{-35}$,
$4.872151964\ 10^{-34}$, $4.979504465\ 10^{-33}$, $4.929709422\ 10^{-32}$,
$4.727672150\ 10^{-31}$, $4.392159933\ 10^{-30}$, $3.952943941\ 10^{-29}$,
$3.446472998\ 10^{-28}$, $2.910944117\ 10^{-27}$, $2.381681549\ 10^{-26}$,
$1.887571497\ 10^{-25}$, $1.448988708\ 10^{-24}$, $1.077291605\ 10^{-23}$,
$7.756499554\ 10^{-23}$, $5.407700392\ 10^{-22}$, $3.650197765\ 10^{-21}$,
$2.385129224\ 10^{-20}$, $1.508433077\ 10^{-19}$, $9.231610431\ 10^{-19}$,
$5.466085124\ 10^{-18}$, $3.130576025\ 10^{-17}$, $1.733857491\ 10^{-16}$,
$9.283819223\ 10^{-16}$, $4.804376448\ 10^{-15}$, $2.402188224\ 10^{-14}$,
$1.160081142\ 10^{-13}$, $5.409053038\ 10^{-13}$, $2.434073867\ 10^{-12}$,
$1.056674420\ 10^{-11}$, $4.423288268\ 10^{-11}$, $1.784568025\ 10^{-10}$,
$6.935480283\ 10^{-10}$, $2.594960600\ 10^{-9}$, $9.341858163\ 10^{-9}$,
$3.233720132\ 10^{-8}$, $1.075563435\ 10^{-7}$, $3.434863874\ 10^{-7}$,
0.000001052383825, 0.000003090685129, 0.000008692551926,
0.00002338923765, 0.00006014375398, 0.0001476255779,
0.0003454438523, 0.0007695531364, 0.001629641936,
0.003275105637, 0.006235297272, 0.01122353509, 0.01905883318,
0.03045850910, 0.04568776362, 0.06413053063, 0.08395269461,
0.1021046286, 0.1148677071, 0.1189338207, 0.1126741459,
0.09699774300, 0.07525686955, 0.05210090970, 0.03179038558,
0.01683020413, 0.007573591859, 0.002816625071, 0.0008311352671,
0.0001824443269, 0.00002648385390, 0.000001906837481]

A várható érték és a szórás is számítható közvetlenül, Maple függvénnyel:

```
> varhato := ExpectedValue(Y);
```

(2.2.2.1.13)

$$varhato := 112.5 \tag{2.2.2.1.13}$$

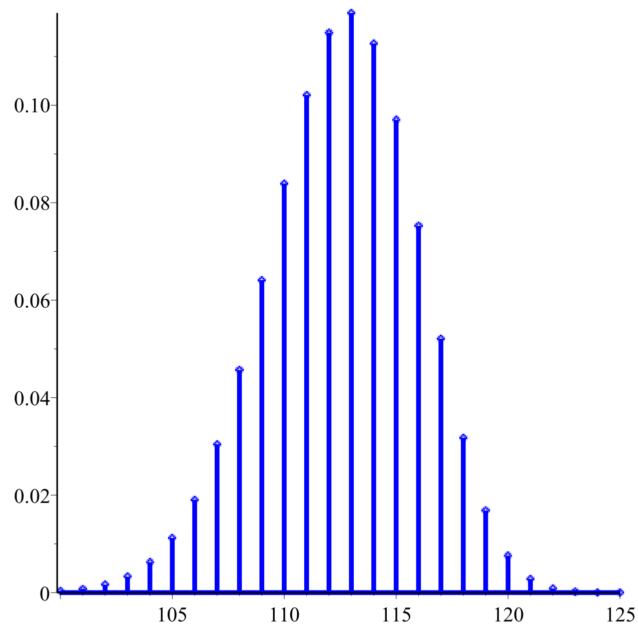
```
> szoras := StandardDeviation(Y);
```

$$szoras := 3.354101966 \tag{2.2.2.1.14}$$

c) Ábrázoljuk az eloszlást pálcikadiagramon (a tartomány megfelel megválasztása mellett)!

A várható érték közelében csoportosulnak a legnagyobb valószínségek, válasszunk ennek megfelel szkítést az ábrázoláshoz, pl. a 100-125 tartományt!

```
> with(DynamicSystems):
Warning, the global variable(s) {y} used by DynamicSystems are assigned values.  They must be unassigned to load DynamicSystems.  DynamicSystems:-SystemOptions may be used to reassign the options that use these variable(s):
  outputvariable = y
> DiscretePlot(y[101..126], py[101..126], style = stem, color = blue, thickness = 3); # indexek 1-gyel eltolva
```



Jól látszik, hogy a maximális valószínség a 113 értéknél van, történetesen a várható érték közelében. Az is jól kivehet, hogy az öt legmagasabb érték majdnem mindegyike 10 % felett van, tehát az esetek b felében az utaslétszám 111 és 115 közé esik. Ez pontosan ki is számítható:

```
> sum(py[i], i = 112..116); # az indexek azért vannak 1-
```

```
gyel eltolva, mert az 1-es index pozíció tartalmazza a 0-hoz tartozó valószínséget
```
0.5455780453 (2.2.2.1.15)

A többi értékhez tartozó valószínség jóval kisebb (121 további érték osztozik a maradékon), egyre távolodva a maximumhelytl azok rohamosan csökkennek.

d) Átlagosan hány utas nem jelenik meg az indulásnál?

A meg nem jelen utasok várható számát úgy kapjuk, ha az összesbl kivonjuk a megjelenket:

```
> meg_nem_jeleno := n - EY;
```
meg_nem_jeleno := 12.5 (2.2.2.1.16)

e) Mekkora a valószínsége, hogy 125 eladott jegy esetén az összes (indulásig megjelen) utas felfér?

Akkor fér fel mindenki, ha a megjelen utasok száma legfeljebb 120, így a 0-120 közötti értékekhez tartozó valószínségeket kell összeadni:

```
> sum(py[i], i = 1..121);
```
0.9961414046 (2.2.2.1.17)

A gép ezt most gyorsan számolja, de kiélezettebb helyzetben lehet a komplementer esemény valószínségével is számolni, mivel a kimaradók kevesebben vannak:

```
> 1 - (sum(py[i], i = 122..126));
  1/(1 - %);
```
0.9961414046

259.1616628 (2.2.2.1.18)

99.6 % esély van arra, hogy minden utas felfér. Ez azt jelenti, hogy 10 %-os lemondási ráta mellett a 120 személyes gépre bátran eladhatunk rendszeresen 5-tel több jegyet, mert átlagosan kb. 260 járatból csak egyszer fordul el, hogy nem fér fel mindenki!

f) Mennyi a valószínsége, hogy marad üres ülés a járaton?

Akkor marad üres ülés, ha a beszállók száma 120-nál kevesebb. Ez a valószínség:

```
> sum(py[i], i = 1..120);
```
0.9885678127 (2.2.2.1.19)

Vagy az ellentét-eseménnyel számolva:

```
> 1 - (sum(py[i], i = 121..126));
```
0.9885678128 (2.2.2.1.20)

Nagyon valószín tehát, hogy marad üres szék. De kicsivel ennél is nagyobb az a valószínség, hogy mindenki felfér, hiszen utóbbi esemény az elbbit részhalmazként tartalmazza (ha maradt üres szék, akkor biztos felfért mindenki). A két esemény közti különbség az, amikor épp 120-an szállnak fel, tehát mindenki felfért, de nem maradt üres szék. Ha tehát az üres szék valószínségéhez hozzáadjuk a pontosan 120 utas valószínségét, akkor megkapjuk a "mindenki felfér" valószínséget:

```
> % + py[121];
```
0.9961414047 (2.2.2.1.21)

3. Feladat (Út forgalma)

Egy útszakasz egy pontján megfigyelések szerint óránként átlagosan 360 járm halad át.
a) Mennyi járm halad át átlagosan 1 perc alatt?
b) Mennyi a valószínsége, hogy 1 perces megfigyelési idtartam alatt éppen 10 járm halad át?
c) Ábrázoljuk pálcikadiagramon az 1 perc alatt áthaladó jármvek darabszámának valószínségi eloszlását egy észszer tartományon!
d) Mennyi járm halad át átlagosan negyed óra alatt?

e) Mennyi a valószínsége, hogy negyed óra alatt éppen 100 járm halad át?
f) Ábrázoljuk pálcikadiagramon a negyed óra alatt áthaladó jármvek darabszámának valószínségi eloszlását egy észszer tartományon!
g) Mennyi a valószínsége, hogy a jármvek negyed óra alatt mérhet száma az átlagos 90-tl több mint 20-szal tér el?

Megoldás

```
> restart;
```

a) Mennyi járm halad át átlagosan 1 perc alatt?
Az átlagos megfigyelt darabszám arányos az idvel, tehát egy perc alatt arányosan kevesebb:

```
> lambda_ora := 360; lambda_perc := lambda_ora/60;
```

$$lambda\_ora := 360$$
$$lambda\_perc := 6 \quad (2.2.3.1.1)$$

Tehát 1 perc alatt átlagosan 6 járm halad át, vagyis átlagosan 10 másodpercenként 1 járm.
b) Mennyi a valószínsége, hogy 1 perces megfigyelési idtartam alatt éppen 10 járm halad át?
Az 1 perc alatt megszámolható jármvek száma (az eredeti megfigyeléssel megegyez körülmények mellett) Poisson-eloszlást követ:

```
> poisson_eloszlas := (lambda, k) -> lambda^k/k!*exp(-lambda);
```

$$poisson\_eloszlas := (\lambda, k) \to \frac{\lambda^k e^{-\lambda}}{k!} \quad (2.2.3.1.2)$$

A Poisson-eloszlástól kell megkérdezni, hogy milyen valószínséggel figyelhetünk meg 1 perc alatt pontosan 10 jármvet:

```
> poisson_eloszlas(lambda_perc, 10); evalf(%);
```

$$\frac{2916}{175} e^{-6}$$
$$0.04130309341 \quad (2.2.3.1.3)$$

Tehát kb. 4 % az esélye, hogy egy perc alatt épp 10 járm halad át.
c) Ábrázoljuk pálcikadiagramon az 1 perc alatt áthaladó jármvek darabszámának valószínségi eloszlását egy észszer tartományon!
A függvény felhasználásával könnyen legyárthatók az eloszlás elemei egy adott darabszámig:

```
> maxdb := 15; # úgy kell megválasztani, hogy az összes
  jelentsebb valószínség ráférjen a diagramra
```

$$maxdb := 15 \quad (2.2.3.1.4)$$

```
> y := [seq(i, i = 0..maxdb)];
```

$$y := [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] \quad (2.2.3.1.5)$$

```
> py := [seq(poisson_eloszlas(lambda_perc, i), i = 0..
  maxdb)]; evalf(%);
```

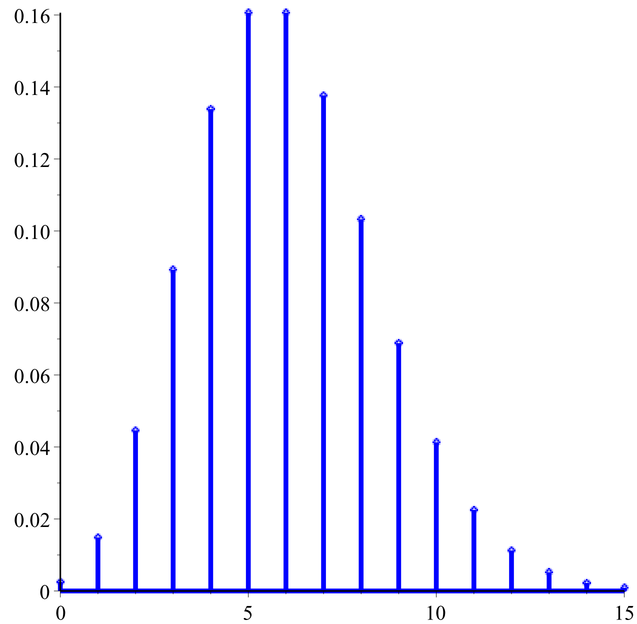
$$py := \left[e^{-6}, 6\,e^{-6}, 18\,e^{-6}, 36\,e^{-6}, 54\,e^{-6}, \frac{324}{5} e^{-6}, \frac{324}{5} e^{-6}, \frac{1944}{35} e^{-6}, \frac{1458}{35} e^{-6}, \frac{972}{35} e^{-6}, \frac{2916}{175} e^{-6}, \frac{17496}{1925} e^{-6}, \frac{8748}{1925} e^{-6}, \frac{52488}{25025} e^{-6}, \frac{157464}{175175} e^{-6}, \frac{314928}{875875} e^{-6}\right]$$

$$[0.002478752177, 0.01487251306, 0.04461753919, 0.08923507837, \quad (2.2.3.1.6)$$

0.1338526176, 0.1606231411, 0.1606231411, 0.1376769781,
0.1032577336, 0.06883848903, 0.04130309341, 0.02252896005,
0.01126448002, 0.005198990779, 0.002228138906, 0.0008912555622]

```
> with(DynamicSystems):
  DiscretePlot(y, evalf(py), style = stem, color = blue,
  thickness = 3); # az evalf() hívás azért kell, mert py
  nem numerikus típusú, hanem algebrai kifejezések listája
```

Warning, the global variable(s) {y} used by DynamicSystems are assigned values. They must be unassigned to load DynamicSystems. DynamicSystems:-SystemOptions may be used to reassign the options that use these variable(s):
  outputvariable = y



Vegyük észre, hogy a percenkénti 6-os átlagérték maga is csak az esetek mintegy 16 %-ában következik be, és mégcsak nem is abszolút leggyakoribbként: történetesen ugyanannyiszor, ahányszor az 5-ös.

d) Mennyi járm halad át átlagosan negyed óra alatt?

Ha a vizsgálatainkat egy perc helyett negyed órára bvítjük, ez idarányosan más várható értéket jelent:

```
> lambda_negyedora := lambda_ora/4;
```

$$lambda\_negyedora := 90 \quad \textbf{(2.2.3.1.7)}$$

e) Mennyi a valószínsége, hogy negyed óra alatt éppen 100 járm halad át?
A negyed órás idtartamra vonatkozó darabszámok eloszlását a negyed órára vetített *lambda* paraméterrel felírt Poisson-eloszlás adja. Ezzel a paraméterrel kell rákérdezni az adott darabszám valószínségére:

```
> poisson_eloszlas(lambda_negyedora, 100): evalf(%);
```

$$0.02332082543 \tag{2.2.3.1.8}$$

Tehát 2.3 % esélye van, hogy 100 autó érkezzen negyed óra alatt.

f) Ábrázoljuk pálcikadiagramon a negyed óra alatt áthaladó jármvek darabszámának valószínségi eloszlását egy észszer tartományon!
A várható érték (90) egy környezetére számoljuk ki az eloszlást:

```
> mindb := 60; maxdb := 120;
```

$$mindb := 60$$
$$maxdb := 120 \tag{2.2.3.1.9}$$

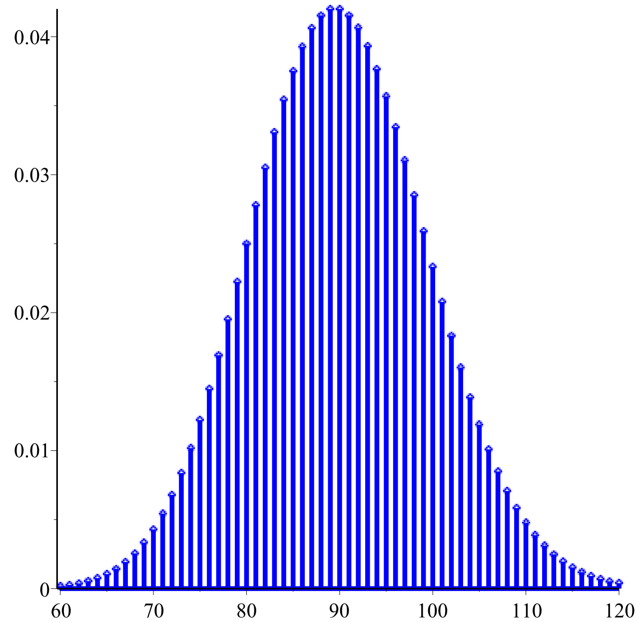
```
> y := [seq(i, i = mindb..maxdb)];
```

$$y := [60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120] \tag{2.2.3.1.10}$$

```
> py := evalf([seq(poisson_eloszlas(lambda_negyedora, i),
  i = mindb..maxdb)]);
```

$$py := [0.0001769588738, 0.0002610868630, 0.0003789970591, 0.0005414243702, 0.0007613780206, 0.001054215721, 0.001437566892, 0.001931060004, 0.002555814712, 0.003333671363, 0.004286148896, 0.005433146487, 0.006791433109, 0.008372999719, 0.01018337804, 0.01222005365, 0.01447111616, 0.01691429162, 0.01951649033, 0.02223397632, 0.02501322337, 0.02779247041, 0.03050393093, 0.03307655162, 0.03543916245, 0.03752381906, 0.03926911297, 0.04062322032, 0.04154647532, 0.04201328965, 0.04201328965, 0.04155160515, 0.04064830938, 0.03933707360, 0.03766315558, 0.03568088423, 0.03345082896, 0.03103685162, 0.02850323107, 0.02591202825, 0.02332082543, 0.02078093355, 0.01833611783, 0.01602185054, 0.01386506297, 0.01188433968, 0.01009047709, 0.008487317175, 0.007072764315, 0.005839897141, 0.004778097661, 0.003874133239, 0.003113142781, 0.002479494250, 0.001957495461, 0.001531952969, 0.001188584201, 0.0009142955384, 0.0006973440549, 0.0005274030668, 0.0003955523001] \tag{2.2.3.1.11}$$

```
> DiscretePlot(y, py, style = stem, color = blue,
  thickness = 3);
```



Látjuk a diagramon, hogy maga az átlagos érték is csak az esetek kb. 4 %-ában szerepel, mivel a valószínségen osztoznia kell sok, hozzá közeli, nála nem sokkal kevésbé valószín értékkel is (nem beszélve a távolabbiakról). Az átlagtól jobban eltér értékek viszont egyre kevésbé részesednek az esélyekbl: a 90-es átlagtól 20-nál nagyobb eltérés már elég ritkán, 30-nál nagyobb eltérés szinte soha nem fordul el.

g) Mennyi a valószínsége, hogy a jármvek negyed óra alatt mérhet száma az átlagos 90-tl több mint 20-szal tér el?

A fenti diagramon szépen látszik, hogy az eloszlás nagy része a 90-es átlag 20 sugarú (70-tl 110-ig terjed) környezetében van. Erre a tartományra összegezve a Poisson-eloszlás értékeit:

```
> `P(70 <= X <= 110)` := evalf(sum(poisson_eloszlas
  (lambda_negyedora, i), i = 70..110));
```

$$P(70 <= X <= 110) := 0.9694992315 \quad (2.2.3.1.12)$$

Ezek szerint nagyjából 97 % eséllyel az észlelt autók száma a [70, 110] intervallumba esik. Kérdésünk ennek komplementerére vonatkozik:

```
> `P(X < 70 or 110 < X)` := evalf(1 - `P(70 <= X <= 110)`)
  ;
```

$$P(X < 70 \text{ or } 110 < X) := 0.0305007685 \quad (2.2.3.1.13)$$

Tehát nagyjából 3 % eséllyel tér el a negyed óra alatt észlelt jármvek száma az átlagos 90-tl 20-nál többel.

4. Feladat (Selejtes alkatrészek)

Egy üzem által gyártott alkatrészek 2 %-a selejtes. A megrendel 10000 db-os csomagban kapja az alkatrészeket. Jelölje X a csomagban található selejtek számát! Válaszoljunk az alábbi kérdésekre!
a) Mi a csomagban található selejtek számának eloszlása?
b) Mi a selejtes darabok számának várható értéke és szórása? Ábrázoljuk az eloszlást egy észszer tartományon!
c) Legalább hány alkatrészt kell véletlenszeren kivenni és megvizsgálni ahhoz, hogy legalább 0.96 valószínséggel legyen köztük selejtes is? (A kiválasztott darabokat vizsgálat után azonnal visszatesszük.)

Megoldás

```
> restart;
```

a) Mi a csomagban található selejtek számának eloszlása?

Rögzítsük az adatokat!

```
> p := 0.02; # selejtarány
  n := 10000; # alkatrészek száma
```

$$p := 0.02$$
$$n := 10000 \qquad (2.2.4.1.1)$$

$n = 10000$ alkatrész van a csomagban, melyek egymástól függetlenül, p valószínséggel selejtesek (ennyi annak a valószínsége, hogy egy véletlenszeren kiválasztott alkatrész selejtes). A selejtek számát ekkor az (n, p)-paraméter binomiális eloszlás írja le: $X \sim Binomial(n, p)$.

b) Mi a selejtes darabok számának várható értéke és szórása? Ábrázoljuk az eloszlást egy észszer tartományon!

A várható érték (a binomiális eloszlásra tanult képlet alapján):

```
> EX := n*p;
```

$$EX := 200.00 \qquad (2.2.4.1.2)$$

Tehát várhatóan 200 selejtes alkatrész lesz a csomagban.

A szórás (a binomiális eloszlásra tanult képlet alapján):

```
> sigmaX := sqrt(n*p*(1-p));
```

$$sigmaX := 14.00000000 \qquad (2.2.4.1.3)$$

Tehát a selejtek számának szórása $\sigma(X) = 14$.

Az eloszlás, mint függvény:

```
> binomialis := k -> binomial(n, k)*p^k*(1-p)^(n-k);
```

$$binomialis := k \rightarrow \mathrm{binomial}(n, k)\, p^k\, (1-p)^{n-k} \qquad (2.2.4.1.4)$$

Ugyanez a Maple eszközeivel:

```
> with(Statistics):
> X := RandomVariable(Binomial(n, p));
```

$$X := \_R \qquad (2.2.4.1.5)$$

```
> EX := ExpectedValue(X);
```

$$EX := 200.00 \qquad (2.2.4.1.6)$$

```
> sigmaX := StandardDeviation(X);
```

$$sigmaX := 14.00000000 \qquad (2.2.4.1.7)$$

Az eloszlást a Statistics csomag *ProbabilityFunction* eljárásával is megkaphatjuk:

```
> binomialis_alternativ := k -> ProbabilityFunction(X, k);
```

$$binomialis\_alternativ := k \rightarrow Statistics{:-}ProbabilityFunction(X, k) \qquad (2.2.4.1.8)$$

```
> binomialis(200) = binomialis_alternativ(200);
```

```
binomialis(180) = binomialis_alternativ(180);
```
$$0.02848400152 = 0.02848400152$$
$$0.01043802840 = 0.01043802840 \qquad (2.2.4.1.9)$$

Az ábrázolást elég a várható érték egy megfelel környezetében elvégezni. Kísérletezzünk!

```
> binomialis(250)/binomialis(200);
  binomialis(150)/binomialis(200);
```
$$0.002423391525$$
$$0.001076687453 \qquad (2.2.4.1.10)$$

A $[150, 250]$ intervallum megfelelnek tnik, azon kívül a valószínségek a maximális érték 2.5 %-a alatt maradnak. (Általánosságban szokás a 3σ-szabályt alkalmazni (fleg normál eloszlásra, de binomiálisra is), mely szerint a valószínségek zöme a várható érték 3σ-sugarú körzetében csoportosul.)

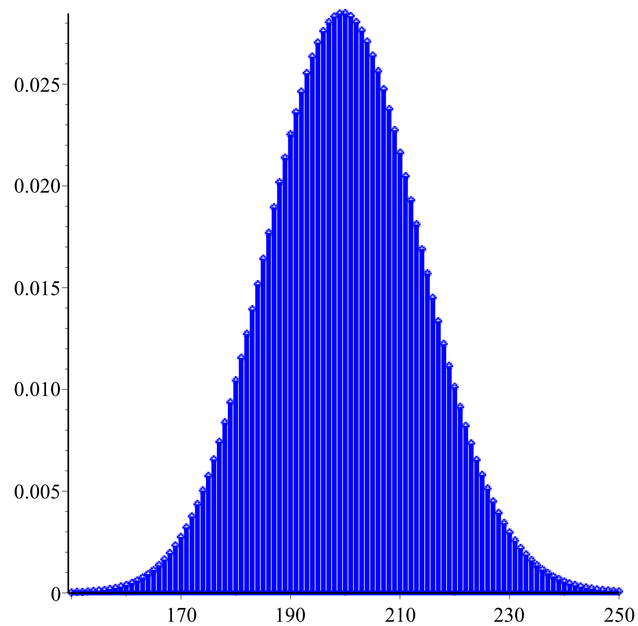
```
> x := [seq(k, k = 150..250)];
```
x := [150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250] **(2.2.4.1.11)**

```
> px := [seq(binomialis(k), k = 150..250)];
```
px := [0.00003066836705, 0.00004082760041, 0.00005398912947, 0.00007091969416, 0.00009254521977, 0.0001199736977, 0.0001545187146, 0.0001977228944, 0.0002513803214, 0.0003175568122, 0.0003986067076, 0.0004971846882, 0.0006162509633, 0.0007590681078, 0.0009291877772, 0.001130425600, 0.001366822692, 0.001642592491, 0.001962051987, 0.002329536908, 0.002749301002, 0.003225400269, 0.003761563745, 0.004361053258, 0.005026515409, 0.005759829786, 0.006561958215, 0.007432800359, 0.008371061444, 0.009374138124, 0.01043802840, 0.01155727128, 0.01272492113, 0.01393256114, 0.01517035854, 0.01642716375, 0.01769065309, 0.01894751385, 0.02018366842, 0.02138453240, 0.02253530047, 0.02362125201, 0.02462806770, 0.02554214742, 0.02635091937, 0.02704313087, 0.02760911060, 0.02804099453, 0.02833290760, 0.02848109530, 0.02848400152, 0.02834229008, 0.02805880992, 0.02763850603, 0.02708827967, 0.02641680315, 0.02563429630, 0.02475227231, 0.02378326164, 0.02274052318, 0.02163775147, 0.02048878875, 0.01930735011, 0.01810676852, 0.01689976574, 0.01569825414, 0.01451317241, 0.01335435708, 0.01223045080, 0.01114884630, 0.01011566472, 0.009135765162, 0.008212782454, 0.007349188870, 0.006546375693, 0.005804750002, 0.005123842451, 0.004502421658, 0.003938611428, 0.003430007207, 0.002973788857, 0.002566827205, 0.002205782456, **(2.2.4.1.12)**

0.001887193048, 0.001607554029, 0.001363384511, 0.001151284136,
0.0009679788435, 0.0008103564954, 0.0006754931355,
0.0005606707903, 0.0004633878326, 0.0003813629497,
0.0003125337755, 0.0002550511915, 0.0002072702561,
0.0001677386218, 0.0001351832206, 0.0001084958813,
0.00008671845218, 0.00006902788787]

```
> DynamicSystems[DiscretePlot](x, px, style = stem, color
  = blue, thickness = 3);
```

Warning, the global variable(s) {x} used by DynamicSystems are assigned values. They must be unassigned to load DynamicSystems. DynamicSystems:-SystemOptions may be used to reassign the options that use these variable(s):
statevariable = x



c) Legalább hány alkatrészt kell véletlenszeren kivenni és megvizsgálni ahhoz, hogy legalább 0.96 valószínséggel legyen köztük selejtes is? (A kiválasztott darabokat vizsgálat után azonnal visszatesszük.)
Vegyünk ki a csomagból m alkatrészt és jelölje Y a köztük lév selejtek számát. Az a) feladatrészhez hasonlóan $Y \sim Binomial(m, p)$. Úgy kell m-et megválasztani, hogy $P(Y \geq 1) \geq 0.96$ teljesüljön! Vegyük észre, hogy

$$P(Y \geq 1) = 1 - P(Y < 1) = 1 - P(Y = 0) = 1 - \binom{m}{0} \cdot p^0 \cdot (1-p)^{m-0} = 1 - (1-p)^m$$

Tehát meg kell oldani az $1 - (1-p)^m \geq 0.96$ exponenciális egyenltlenséget m-re!

```
> solve(1-(1-p)^m >= 0.96);
```

$$RealRange(159.3289342, \infty) \qquad (2.2.4.1.13)$$

Tehát $m \geq 159.329$, és mivel m-nek egésznek kell lennie, ezért a legkisebb megfelel mintaméret az $m = 160$. ($m < n$ teljesül, ezért valóban ki is tudunk venni ennyi alkatrészt a csomagból.)

5. Feladat (Céllövölde)

Egy vásári céllövöldében 3-szor kell eltalálnunk a céltáblát ahhoz, hogy megnyerjük a fdíjat, egy 2500 Ft érték ajándékutalványt. Egy próbálkozás 250 Ft-ba kerül és minden alkalommal 25 %-os eséllyel találunk célba.

a) Jelölje *X* az els három próbálkozásból elért találatok számát! Adjuk meg és ábrázoljuk pálcikadiagramon *X* eloszlását!

b) Határozzuk meg *X* várható értékét, varianciáját és szórását!

c) Jelölje *Y* azt, hogy hányadik próbálkozásra sikerül elször betalálni! Adjuk meg *Y* eloszlását és ábrázoljuk azt egy észszer tartományon!

d) Határozzuk meg *Y* várható értékét, varianciáját és szórását!

e) Mekkora a valószínsége, hogy pont a harmadik próbálkozásra sikerül elször eltalálnunk a célt?

f) Döntsük el, hogy kedvez vagy kedveztlen-e a játék, ha addig próbálkozunk, amíg meg nem nyerjük a fdíjat?

g) Milyen jó céllövnek kellene lennünk, hogy megérje játszani?

Megoldás

```
> restart;
```

a) Jelölje *X* az els három próbálkozásból elért találatok számát! Adjuk meg és ábrázoljuk pálcikadiagramon *X* eloszlását!

```
> m := 3;
  s := 2500;
  t := 250;
  p := 0.25;
```

$$m := 3$$
$$s := 2500$$
$$t := 250$$
$$p := 0.25 \qquad (2.2.5.1.1)$$

X binomiális eloszlású $n = 3$ és $p = 0.25$ paraméterrel, mert 3 lövésünk van és mindegyik –a többitől függetlenül– 25 %-os valószínséggel sikeres.
Ábrázoljuk az eloszlást pálcikadiagramon!

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

```
> X := RandomVariable(Binomial(3, p));
```

$$X := \_R \qquad (2.2.5.1.2)$$

Az eloszlás:

```
> binomialis := k -> ProbabilityFunction(X, k); # k ->
  binomial(n, k)*p^k*(1 - p)^(n - k)
```

$$binomialis := k \rightarrow Statistics\text{:-}ProbabilityFunction(X, k) \qquad (2.2.5.1.3)$$

```
> x := [seq(k, k = 0..3)];
```

$$x := [0, 1, 2, 3] \quad (2.2.5.1.4)$$

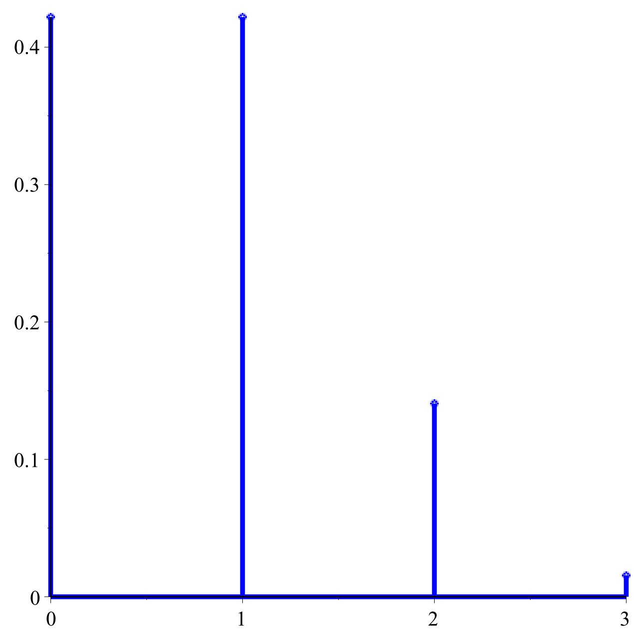
```
> px := [seq(binomialis(k), k = 0..3)];
```

$$px := [0.4218750, 0.421875, 0.140625, 0.0156250] \quad (2.2.5.1.5)$$

Ábrázoljuk az eloszlást pálcikadiagramon.

```
> DynamicSystems[DiscretePlot](x, px, style = stem, color
  = blue, thickness = 3);
```

Warning, the global variable(s) {s, t, x} used by DynamicSystems are assigned values. They must be unassigned to load DynamicSystems. DynamicSystems:-SystemOptions may be used to reassign the options that use these variable(s):
  complexfreqvar = s
  continuoustimevar = t
  statevariable = x



b) Határozzuk meg *X* várható értékét, varianciáját és szórását!

```
> EX := ExpectedValue(X); # = 3*p
```

$$EX := 0.75 \quad (2.2.5.1.6)$$

```
> VarX := Variance(X); # = 3*p*(1-p)
```

$$VarX := 0.5625 \quad (2.2.5.1.7)$$

```
> sigmaX := StandardDeviation(X); # = sqrt(VarX)
```

$$sigmaX := 0.7500000000 \quad (2.2.5.1.8)$$

Tehát X várható értéke 0.75, varianciája 0.5625, szórása pedig 0.75.

c) Jelölje Y azt, hogy hányadik próbálkozásra sikerül elször betalálni! Adjuk meg Y eloszlását és ábrázoljuk azt egy észszer tartományon!

Y eloszlása geometriai p paraméterrel, mert a fent vázolt kísérletsorozatban az els sikeres kísérlet sorszámát adja meg.

Vigyázzunk! A Maple picit máshogy definiálja a geometriai eloszlást, mint ahogyan mi tanuljuk, az értékek el vannak tolva 1-gyel. Ezért biztosabb, ha a tanult képletek alapján számolunk!

```
> geometriai := y -> p*(1 - p)^(y-1);
```

$$geometriai := y \rightarrow p\ (1-p)^{y-1} \qquad \textbf{(2.2.5.1.9)}$$

```
> y := [seq(k, k = 1..20)];
```

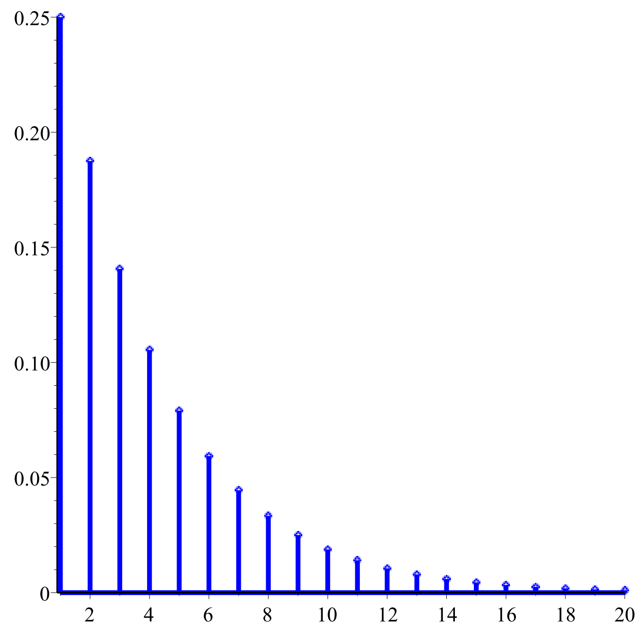
$$y := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] \qquad \textbf{(2.2.5.1.10)}$$

```
> py := [seq(geometriai(k), k = 1..20)];
```

py := [0.250, 0.1875, 0.140625, 0.10546875, 0.0791015625, 0.05932617188, 0.04449462890, 0.03337097168, 0.02502822875, 0.01877117157, 0.01407837868, 0.01055878401, 0.007919088005, 0.005939316005, 0.004454487002, 0.003340865252, 0.002505648940, 0.001879236704, 0.001409427528, 0.001057070646] **(2.2.5.1.11)**

Ábrázoljuk Y eloszlását pálcikadiagramon!

```
> DynamicSystems[DiscretePlot](y, py, style = stem, color
  = blue, thickness = 3);
```



d) Határozzuk meg Y várható értékét, varianciáját és szórását!

```
> EY := 1/p;
```

$$EY := 4.000000000 \quad (2.2.5.1.12)$$

```
> VarY := (1-p)/p^2;
```

$$VarY := 12.00000000 \quad (2.2.5.1.13)$$

```
> sigmaY := sqrt(VarY);
```

$$sigmaY := 3.464101615 \quad (2.2.5.1.14)$$

Tehát várhatóan negyedszerre sikerül elször eltalálni a céltáblát, Y varianciája 12, szórása pedig 3.464.

e) Mekkora a valószínsége, hogy pont a harmadik próbálkozásra sikerül elször eltalálnunk a célt?

A $P(Y=3)$ valószínséget kell meghatározni. Ez a c) részben felírt eloszlással könnyen megtehet:

```
> `P(Y = 3)` := geometriai(3);
```

$$P(Y = 3) := 0.140625 \quad (2.2.5.1.15)$$

Azaz nagyjából 14 %-nyi esélyünk van arra, hogy a harmadik lövésre találjuk el elször a célt.

f) Döntsük el, hogy kedvez vagy kedveztlen-e a játék, ha addig próbálkozunk, amíg meg nem nyerjük a fdíjat?

A várható nyereményre/veszteségre vonatkozik a kérdés. Legyen $Y_1 = Y$ az els találat sorszáma, Y_2 az els találat után a második találatig leadott lövések száma, Y_3 pedig a második találat után a harmadik találatig leadott lövések száma. Ekkor $Z = Y_1 + Y_2 + Y_3$ a nyerésig összesen leadott lövések száma. A játék értéke: $W = 2500 - 250 \cdot Z$, mert 2500 Ft érték utalványt nyerünk és Z-szer (minden próbálkozásért) fizetünk be 250 Ft-ot. A játék várható értéke az additivitást használva:

$$E(W) = E(2500 - 250 \cdot Z) = 2500 - 250 \cdot E(Z)$$

Vegyük észre, hogy Y_i-k azonos eloszlásúak, mert egy találat után a játék "újraindul", ugyanúgy 25 % valószínséggel találunk be minden próbálkozásra, és az els (következ) sikeres találat sorszámát nézzük (nem javul a célzóképességünk). Emiatt $Y = Y_1$-hez hasonlóan Y_2 és Y_3 is p paraméter geometriai eloszlású és $E(Z) = E(Y_1 + Y_2 + Y_3) = E(Y_1) + E(Y_2) + E(Y_3) = 3 \cdot E(Y)$.
(ismét kihasználtuk a várható érték additivitását).

```
> EY1 := EY;
  EY2 := EY;
  EY3 := EY;
```

$$EY1 := 4.000000000$$
$$EY2 := 4.000000000$$
$$EY3 := 4.000000000 \qquad (2.2.5.1.16)$$

```
> EZ := EY1 + EY2 + EY3;
```

$$EZ := 12.00000000 \qquad (2.2.5.1.17)$$

```
> EW := 2500 - 250*EZ;
```

$$EW := -500.000000 \qquad (2.2.5.1.18)$$

Tehát várhatóan 500 Ft-ot *veszítünk*, a játék kedveztlen!
g) Milyen jó céllövnek kellene lennünk, hogy megérje játszani?
Tegyük fel, hogy p helyett p' az esélye annak, hogy egy próbálkozásunk sikeres. Ekkor Y p' paraméter geometriai eloszlású, várható értéke $E(Y) = \frac{1}{p'}$. Az f) feladatrész alapján $E(W) = 2500 - 250 \cdot 3 \cdot E(Y) = 2500 - \frac{750}{p'}$, és a játék kedvezségéhez $E(W) > 0$-nak kell teljesülnie.

```
> solve(0 < 2500 - 750/`p'`);
```

$$RealRange(-\infty, Open(0)), RealRange\left(Open\left(\frac{3}{10}\right), \infty\right) \qquad (2.2.5.1.19)$$

Vagyis $p' > \frac{3}{10}$. Eszerint több, mint 30 %-os eséllyel kell célba találnunk ahhoz, hogy megérje játszani.

6. Feladat (Telefonos ügyfélszolgálat)

Egy cégnek 1000 ügyfele van és minden ügyfél azonos eséllyel hívja fel a telefonos ügyfélszolgálatot a nap egy adott idszakában (pl. 15:00 és 16:00 között). Tudjuk továbbá, hogy az esetek huszadrészében egyáltalán nem fut be hívás.
a) Adjuk meg a vizsgált idszakban befutó telefonhívások eloszlását! Mekkora az esélye, hogy legalább 5 hívás történik?
b) Határozzuk meg a hívások számának várható értékét és szórását!
c) Közelítsük a hívások számát Poisson-eloszlással! Mennyi ennek várható értéke és szórása?
d) Ábrázoljuk közös oszlopdiagramon az eredeti és a közelít eloszlást egy észszer

tartományon!
e) Legfeljebb mekkora hibát vétünk a közelítéssel, ha pontosan *k* db hívás beérkezésének valószínségét számoljuk?

Megoldás

```
> restart;
```

a) Adjuk meg a vizsgált idszakban befutó telefonhívások eloszlását! Mekkora az esélye, hogy legalább 5 hívás történik?

Jelölje X az ügyfélszolgálatra befutó hívások számát a vizsgált idszakban! Felfoghatjuk úgy, hogy 1000 független, azonos kísérletet végzünk, ahol egy kísérlet sikeres, ha az adott ügyfél felhívja az ügyfélszolgálatot a vizsgált idszakban. Minden ilyen kísérlet azonos p valószínséggel sikeres. Ezt a konstrukciót a binomiális eloszlás írja le, melynek paraméterei $n = 1000$ és p. A p értékét abból a feltételbl tudjuk meghatározni, hogy az esetek huszadrészében egyáltalán nincs bejöv hívás: $P(X=0) = \frac{1}{20}$, másrészt pedig

$P(X=0) = \binom{n}{0} \cdot p^0 \cdot (1-p)^{n-0} = (1-p)^n$.

```
> n:= 1000;
```

$$n := 1000 \quad \textbf{(2.2.6.1.1)}$$

```
> p := fsolve((1-p)^1000 = 1/20, p, 0..1); # a 3. paraméter a 0 <= p <= 1 tartományra szkíti a keresést
```

$$p := 0.002991249545 \quad \textbf{(2.2.6.1.2)}$$

Tehát nagyjából 0.3 % eséllyel fog egy adott ügyfél betelefonálni. Az eloszlás:
$X \sim Binomial(n, p) \approx Binomial(1000, 0.003)$.
Megadjuk az eloszlást függvényként.

```
> with(Statistics):
> X := RandomVariable(Binomial(n, p));
```

$$X := \_R \quad \textbf{(2.2.6.1.3)}$$

```
> binomialis := k -> ProbabilityFunction(X, k); # k -> binomial(n, k)*p^k*(1-p)^(n-k)
```

$$binomialis := k \rightarrow Statistics\text{:-}ProbabilityFunction(X, k) \quad \textbf{(2.2.6.1.4)}$$

Határozzuk meg a $P(X \geq 5)$ valószínséget az ellentét-eseménnyel számolva!

$$P(X \geq 5) = 1 - P(X < 5)$$

```
> `P(X >= 5)` := 1 - sum(binomialis(k), k = 0..4);
```

$$P(X >= 5) := 0.1830150574 \quad \textbf{(2.2.6.1.5)}$$

Nagyjából 18.3 % az esélye, hogy van legalább 5 bejöv hívás a vizsgált idszakban.

b) Határozzuk meg a hívások számának várható értékét és szórását!

Számolhatunk a binomiális eloszlásra tanult képletekkel is, de használjuk most a Statistics csomag függvényeit.

```
> EX := ExpectedValue(X); # = n*p
```

$$EX := 2.991249545 \quad \textbf{(2.2.6.1.6)}$$

Azaz közel 3 telefonhívásra lehet számítani.

```
> sigmaX := StandardDeviation(X); # = sqrt(n*p*(1 - p))
```

$$sigmaX := 1.726934269 \quad \textbf{(2.2.6.1.7)}$$

c) Közelítsük a hívások számát Poisson-eloszlással! Mennyi ennek várható értéke és szórása?

Ha n nagy és p kicsi, akkor a binomiális eloszlást közelíthetjük $\lambda = n \cdot p$ paraméter Poisson-eloszlással.

```
> lambda := n*p; # = EX
```
$$\lambda := 2.991249545 \quad (2.2.6.1.8)$$

A Poisson-közelítésnél azt feltételezzük, hogy végtelen sok ügyfél végtelenül kicsi valószínséggel hívja az ügyfélszolgálatot, de a hívások átlagos srsége $\lambda > 0$. Hozzuk létre az ehhez tartozó valószínségi változót!

```
> Y := RandomVariable(Poisson(lambda));
```
$$Y := \_R0 \quad (2.2.6.1.9)$$

A várható érték az eloszlás paramétere:

```
> EY := ExpectedValue(Y); # = lambda
```
$$EY := 2.991249545 \quad (2.2.6.1.10)$$

A szórás a paraméter négyzetgyöke:

```
> sigmaY := StandardDeviation(Y); # = sqrt(lambda)
```
$$sigmaY := 1.729522924 \quad (2.2.6.1.11)$$

Vegyük észre, hogy $\sigma(X)$ és $\sigma(Y)$ majdnem megegyezik, bár van egy kis eltérés köztük.

d) Ábrázoljuk közös oszlopdiagramon az eredeti és a közelít eloszlást egy észszer tartományon!

Y eloszlása (függvényként megadva):

```
> poisson_eloszlas := k -> ProbabilityFunction(Y, k); # k
  -> lambda^k/k!*exp(-lambda)
```
$$poisson\_eloszlas := k \rightarrow Statistics:\text{-}ProbabilityFunction(Y, k) \quad (2.2.6.1.12)$$

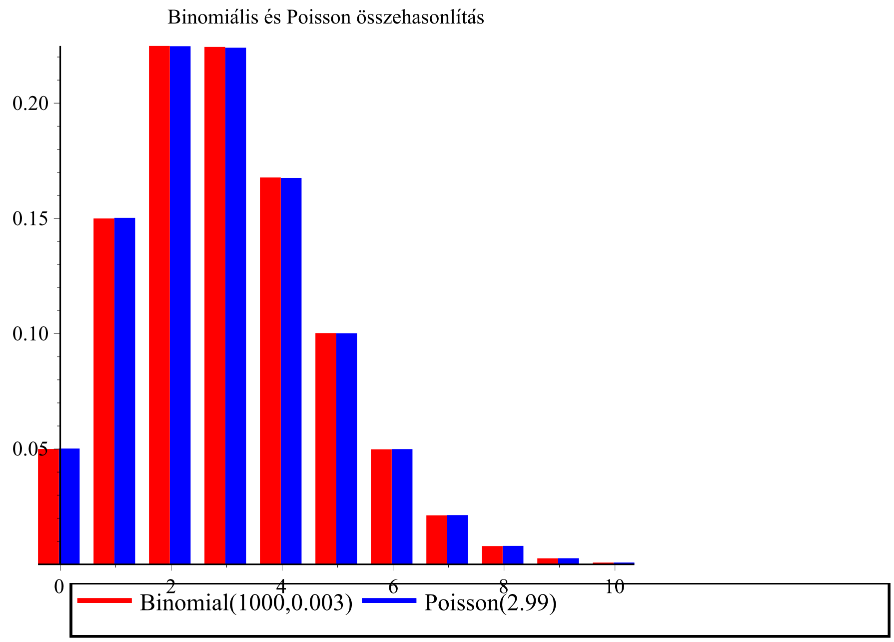
Az eloszlásokat összehasonlító oszlopdiagram elkészítéséhez elbb rögzítsük listában az X lehetséges értékeihez tartozó valószínségeket, majd tegyünk ugyanígy Y-nal, de itt szorítkozzunk csak a 0-1000 tartományra!

```
> px := [seq(binomialis(i), i = 0..1000)]:
> py := [seq(poisson_eloszlas(i), i = 0..1000)]:
```

Az ábrázoláshoz használjuk a Statistics csomag *ColumnGraph* eljárását! A tartomány legyen a 0-10 számok halmaza (ezen kívül már elhanyagolhatók a valószínségek, mint azt az alábbi számítás mutatja)!

```
> binomialis(11)/binomialis(2); # maximumhoz viszonyított
  valószínség
  poisson_eloszlas(11)/poisson_eloszlas(2);
```
$$0.0009348062996$$
$$0.0009606121775 \quad (2.2.6.1.13)$$

```
> ColumnGraph([px[1..11], py[1..11]], title = "Binomiális
  és Poisson összehasonlítás", legend = ["Binomial(1000,
  0.003)", "Poisson(2.99)"], color = [red, blue], offset =
  -0.4); # ha több adatsort ábrázolunk, mindig készítsünk
  "legend"-et!
```



Láthatjuk, hogy szinte észrevehetetlen a különbség az elméletileg precíz eloszlás (binomiális) és a közelítés (Poisson) között! Ilyen esetben érdemes a Poisson-közelítést használni, mert annak valószínűségeit lényegesen gazdaságosabb kiszámítani azonos numerikus pontosság megkövetelése mellett.

e) Legfeljebb mekkora hibát vétünk a közelítéssel, ha pontosan *k* db hívás beérkezésének valószínűségét számoljuk?

A feladat a $|P(X=k) - P(Y=k)|$ eltérések maximumának meghatározása, ahol $k = 0, 1, 2, ...$

A legnagyobb hiba $0 \le k \le 1000$ esetén:

```
> hiba_A := max(seq(abs(px[k + 1] - py[k + 1]), k = 0..1000));
```

$$hiba\_A := 0.0003367616 \qquad \textbf{(2.2.6.1.14)}$$

1000-nél nagyobb értéket X már nem vehet fel, így $k > 1000$ esetén $|P(X=k) - P(Y=k)| = P(Y=k)$. Ez $k = 1001$-ben veszi fel a maximumát, mert $P(Y=k)$ ezen a tartományon szigorúan monoton csökken (igazából már a maximumától, $k_0 = \lfloor \lambda \rfloor + 1$ kezdve csökken).

```
> hiba_B := poisson_eloszlas(n + 1);
```

$$hiba\_B := 2.656664639 \; 10^{-2096} \qquad \textbf{(2.2.6.1.15)}$$

Ez a hiba extrém kicsi.

```
> maxhiba := max(hiba_A, hiba_B);
```

(2.2.6.1.16)

$maxhiba := 0.0003367616$ **(2.2.6.1.16)**

Tehát legfeljebb kb. 0.03 %-os hibát vétünk ha binomiális helyett Poisson eloszlással számolunk. Ez a legtöbb gyakorlati alkalmazásban megengedhet.

Gyakorló feladatok

Gy/1. Feladat (Feleletválasztós teszt)

Vizsgán egy 15 kérdésbl álló feleletválasztós teszt minden kérdésére négy válasz közül lehet választani, de csak egy helyes közülük. A hallgató nem tanult semmit, így csak találgatni tud.
a) Tekintsük a helyes válaszok számát valószínségi változónak. Írjuk fel az értékkészletét, az eloszlását, és szemléltessük pálcikadiagramon!
b) Mennyi a helyes válaszok várható értéke és szórása?
c) Legalább 40 %-os eredmény kell az elégségeshez. Mekkora az esélye, hogy a hallgató átmegy a vizsgán?
d) A hallgatónak a barátai sört fizetnek fájdalomdíjként, ha kiderül, hogy minden válasza rossz. Mi az esély az ingyen sörre?
e) Tegyük fel, hogy a hallgató akárhányszor újra vizsgázhat a tárgyból. Várhatóan hány sikertelen vizsgán lesz túl, mire sikerül megszereznie az aláírást?

Megoldás

```
> restart;
```

Gy/2. Feladat (Repüljegyek eladása / 2.)

Mekkora eséllyel fér fel mindenki egy 2. Feladatban megfogalmazott konstrukcióban, ha a visszamondás esélye 10% helyett csak 5%? Kísérletezzéssel állapítsuk meg, hogy legfeljebb mennyi plusz jegy eladása vállalható ebben az esetben jelentékeny kockázat (1 %) vállalása nélkül?

Megoldás

```
> restart;
```

Gy/3. Feladat (Piros lapok száma)

Egy 32 lapos, alaposan megkevert magyarkártya-csomagot négy játékos között egyenlen osztunk el. Az *X* változó értéke legyen az egyik kijelölt játékoshoz kerül *piros* lapok száma! Készítsük el az *X* valószínségi változó eloszlásának táblázatát, ábrázoljuk azt pálcikadiagramon, továbbá számítsuk ki várható értékét és szórását!

Megoldás

```
> restart;
```

Gy/4. Feladat (Szökevény rab)

Egy szökésben lév rab ellopja a börtönr kulcscsomóját. A kulcscsomón 19 kulcs van, és a rabnak 3 ajtót kell egymás után kinyitnia a szabaduláshoz. Minden ajtót pontosan 1

kulcs nyit, és különböz ajtókhoz különböz kulcsok tartoznak. A rab minden egyes ajtónál véletlenszer sorrendben próbálja ki a kulcsokat addig, amíg meg nem találja a jó kulcsot, utána megy a következ ajtóhoz. Az elz ajtókat nyitó kulcsokat megjegyzi, és nem próbálja újra. Egy kulcs kipróbálása 5 másodpercet vesz igénybe.
a) Határozzuk meg és ábrázoljuk diagramon az els ajtó kinyitásához szükséges *T* id eloszlását!
b) Mekkora az esélye, hogy (legfeljebb) 1 perc alatt átjut az els ajtón?
c) Várhatóan hány percre lesz szüksége összesen a szabaduláshoz?

Megoldás

```
> restart;
```