# Príklad na použitie rozdelení

# náhodnej veličiny

40 zbojníci a ich šéf sa rozhodli vybrať za „zárobkom“ do hlavného mesta. Odchod bol stanovený na 8:15. Niektorí zbojníci sa začali schádzať skôr, iní meškali – dochvíľnosť každého z nich charakterizuje funkcia hustoty pravdepodobnosti

$$f(x) = c*x*e^{-(x/15)} \text{ na } [0, \infty), \text{ kde x sú minúty po ôsmej.}$$

**Úlohy**

a) Doplňte vhodné c.

b) Aká je pravdepdobnosť, že presne o 8:15 sa zíde aspoň polovica z nich (šéfa nepočítame)?

c) Najskôr o koľkej by mal šéf s autobusom vyzdvihnúť tých, čo sa zišli, aby mal s pravdepodobnosťou 80% aspoň 30 chlapov? (odpoveď s presnosťou na celé minúty)

d) S akým počtom chlapov môže veliteľ na 90% počítať, ak príde o 8:30?

e) Aj veliteľ je len zbojník a aj jeho príchod je podriadený tej istej funkcii hustoty ako u kolegov. Je však zaťažený na okrúhle hodnoty času a preto s autobusom počká dovtedy, kým sa na hodinkách objaví údaj o minútach deliteľný piatimi. S akým počtom zbojníkov šéf s najväčšou pravdepodobnosťou odíde?

---

**Riešenia**

a) Dobre definovaná funkcia hustoty musí mať integrál na svojom definičnom intervale rovný jednej. Pozrime sa na hodnotu integrálu funkcie f bez konštanty c.

```
>> f = inline('x.*exp(-x/15)')
```

Primitívnu funkciu k f nájdeme ručne metódou per partes (…vari s tým máme nejaký problém?) alebo požiadame Matlab, aby nám pomohol. Na analytické integrovanie musíme definovať symbolickú premennú s, tj. je to premenná, za ktorú sa nedosadzuje (tak, ako keď na papieri počítame „všeobecne" s premennými bez dosadzovania).

```
>> syms s;      % definícia symbolickej premennej s
>> F=int(f(s))  % integrovanie funkcie f(s)

        F = -15*s*exp(-1/15*s)-225*exp(-1/15*s)

>> s=0; subs(F) % dosadenie 0 do F
        ans = -225
```

Vzhľadom na obor možných udalostí sa zaoberáme integrálom od 0 po ∞. Platí:

$$\int_{[0,\infty)} f(x)\,dx = \lim_{x\to\infty} (F(x)-F(0)) =$$
$$= \lim_{x\to\infty} (-15*x*\exp(-1/15*x)-225*\exp(-1/15*x)+225) = 225.$$

Hľadanou konštantou c je teda číslo 1/225.
Opravíme podľa toho zadanie f a F:

```
>> f = inline('x.*exp(-x/15)/225');
>> F=inline('-x*exp(-1/15*x)/15-exp(-1/15*x)+1')
```

---

b) Najprv sa budeme zaoberať jedným ľubovoľným zbojníkom a jeho osobnej pravdepodobnosti príchodu do 8:15. Tá je daná hodnotou F(15):

```
>> p=F(15)
                p = 0.26424111007043
```

Aspoň polovica pri počte 40 znamená čosi medzi 20 a 40 prítomnými. Pravdepodobnosť včasného príchodu 40, 39, 38, …,20 ľudí získame postupne v cykle z binomického rozdelenia. Tieto hodnoty nakoniec sčítame:

```
>> q=p^40; S=q; for i=39:-1:20, q=q*(i+1)/(40-i)/p*(1-p); S=S+q; end, S

                S = 0.00122077156284
```

Výsledok nevyzerá pre zbojníkov veľmi perspektívne...

---

c) Hľadáme výsledok v celých minútach, tak budeme skúšať. Ako sme videli, 15 minút je málo, skúsme teda 30, 70, 40… posledné naše pokusy boli:

```
>> p=F(45); q=p^40; S=q; for i=39:-1:30, q=q*(i+1)/(40-i)/p*(1-p); S=S+q; end, S
S = 0.8426

>> p=F(44); q=p^40; S=q; for i=39:-1:30, q=q*(i+1)/(40-i)/p*(1-p); S=S+q; end, S
S = 0.7993
```

Ak rátame celé minúty a chceme mať istotu o 80% spoľahlivosti, bude treba počkať 45 minút. Inak by ale na 80% stačilo počkať 44 minút a jednu sekundu:

```
>> p=F(44+1/60); q=p^40; S=q; for i=39:-1:30, q=q*(i+1)/(40-i)/p*(1-p); S=S+q; end, S
        S =   0.80007703427593
```

---

Z pohľadu človeka, ktorý hľadá univerzálne metódy aplikovateľné poľahky aj na iné dané hodnoty, by sa žiadalo nájsť postup, v ktorom „skusmo“ nebude mať až taký podiel.

Vieme, že 15 minút je málo a skusmo si zistíme, že napr. 70 už je zbytočne veľa. Metódu „skusmo“ sme tu obmedzili na minimum – len na nájdenie nejakých časových hraníc, medzi ktorými bude hľadaný výsledok.

Vyskúšame po celých minútach všetky hodnoty v hraniciach 15 až 100. Vyhneme sa ďalším cyklom a využijeme vektorové vymoženosti Matlabu. Najprv si musíme upraviť (=doplniť bodky) vzorec na výpočet hodnôt F tak, aby doň bolo možné dosadzovať vektory – Matlab to spraví za nás:

```
>> F=vectorize(F)
                    F =     Inline function:
                             F(x) = -x.*exp(-1./15.*x)./15-exp(-1./15.*x)+1
```

Použijeme rovnaký cyklus ako pri „skúšaní“. Miesto postupného dosadzovania rôznych p do F však dosadíme všetky skúmané p naraz (v cykle doplníme bodky):

```
>> w=(15:70)'; p=F(w);
>> q=p.^40; S=q; for i=39:-1:30, q=q*(i+1)/(40-i)./p.*(1-p); S=S+q; end
```

Výsledok S si nedáme vypísať. Pri iných zadaných číslach by totiž výpis mohol byť príliš dlhý na to, aby sme sa s ním trápili my. Potrápi sa Matlab:

– Opýtame sa, či sú jednotlivé zložky vektora S väčšie ako 0.8. Odpoveďou bude vektor 0 a 1 (0=nie, 1=áno), ktorý si nebudeme vypisovať.[1]

```
>> h = S>0.8;
```

---

[1] V tomto príklade však odporúčame pozrieť si S aj spomínaný vektor odpovedí v array editore, aby sme mali predstavu o tom, čo a ako sa počíta.

– Vieme, že hodnoty v S postupne narastajú. Vektor h teda samé nuly a za nimi samé jednotky. Opýtame sa, na ktorej pozícii je prvá jednotka:

```
>> i=min(find(h))
                    i = 31
```

Príkaz find vypíše čísla tých pozícií vo vektore, ktoré sú nenulové. Príkaz min vypíše najnižšiu z nich. V našom prípade je to 31. Čo je na 31. pozícii vo vektore w a S?

```
>> [w(i),S(i)]
          ans =  45.00          0.84264275676165
```

Pri čakaní 45 minút je pravdepodobnosť viac ako 80%. 44 minút však nestačí.

Použitím toho istého cyklu skúsime výsledok spresniť. Preskúmame pravdepodobnosti pre čas medzi 44 a 45 minútami krokovaný po sekundách:

```
>> w=(44:1/60:45)'; p=F(w);
>> q=p.^40; S=q; for i=39:-1:30, q=q*(i+1)/(40-i)./p.*(1-p); S=S+q; end
>> h = S>0.8; i=min(find(h)); [w(i),S(i)]

          ans =  44.01666666666667   0.80007720000573
```

Už 1 sekunda nad 44 minút stačí na dosiahnutie potrebnej pravdepodobnosti.

---

d) O 8:30 má každý zo zbojníkov osobnú pravdepodobnosť príchodu:

```
>> p=F(30)
          p = 0.59399415029016
```

Pravdepodobnosti, že o 8:30 príde 40, 39, …, 0 zbojníkov, určuje binomické rozdelenie, a tieto hodnoty si uložíme do vektora:

```
>> q=p^40; S=q; for i=39:-1:0, q=q*(i+1)/(40-i)/p*(1-p); S=[S,q]; end, S
```

Kumulatívna suma vektora S určuje pravdepodobnosť, že o 8:30 sa zíde **aspoň** 40, 39, 38, … zbojníkov. V záujme prehľadnosti si tieto počty napíšeme nad hodnoty kumulatívnej sumy:

```
>> [40:-1:0;cumsum(S)]
```

Vo výpise (príliš dlhý na to, aby sme ho uvádzali) vidno, že pravdepodobnosť nad 90% sa týka očakávania 20 alebo menej zbojníkov. Na 90% možno počítať s 20 chlapmi.

---

e) Ak príde šéf presne o 8:00, okamžite vyráža. Ak príde medzi 8:00+ a 8:05, odchádza o 8:05, atď. Otázkou teda je, v ktorom päťminútovom intervale je najvyššia pravdepodobnosť jeho príchodu. Charakter funkcie f dáva tušiť, že to bude (8:10,8:15] alebo (8:15, 8:20]. Overme si však viacero intervalov.

```
>> format
>> a=(0:5:55)'; b=(5:5:60)'; [a,b,F(b)-F(a)]

ans =
```

| od | do | pravdepodobnosť |
|---|---|---|
| 0 | 5.0000 | 0.0446 |
| 5.0000 | 10.0000 | 0.0997 |
| 10.0000 | 15.0000 | 0.1199 |
| 15.0000 | 20.0000 | 0.1207 |
| 20.0000 | 25.0000 | 0.1114 |
| 25.0000 | 30.0000 | 0.0977 |
| 30.0000 | 35.0000 | 0.0828 |
| 35.0000 | 40.0000 | 0.0685 |
| 40.0000 | 45.0000 | 0.0556 |
| 45.0000 | 50.0000 | 0.0446 |
| 50.0000 | 55.0000 | 0.0353 |
| 55.0000 | 60.0000 | 0.0277 |

Najpravdepodobnejší, hoci len tesne pred druhým zo spomínaných favoritov, je interval medzi 8:15 a 8:20. Najpravdepodobnejší odchod je o 8:20.

Pravdepodobnosť príchodu jedného zbojníka do 8:20 je

```
>> p=F(20)
          p =   0.38494001106330
```

Aká je pravdepodobnosť príchodu 40, 39, …, 0 zbojníkov (tieto počty uložíme do vektora I) do 8:20, určuje binomické rozdelenie a hodnoty si uložíme do vektora S.

```
>> q=p^40; I=40:-1:0; S=q; for i=39:-1:0, q=q*(i+1)/(40-i)/p*(1-p); S=[q,S]; end
```

Zistíme si, ktorý z počtov 40, 39, ... 0 má najvyššiu pravdepodobnosť:

```
>> m=max(S); i=find(S==m); [I(i), S(i)]

ans =  25.00      0.12833127152382
```

Šéf teda s najväčšou[2] pravdepodobnosťou odíde s 25 zbojníkmi.

[2] „Najväčšia“ neznamená nutne „závratne veľká“.