MI-SPI 2013 – Domácí úkol č. 1

*Vedoucí týmu: Jaroslav Žmolík (zmolijar, **102**)*
*Členové týmu: Jiří Nádvorník (nadvoji1, **107**), Filip Mudruněk (mudrufil, **108**)*
Datum: 17. 4. 2014

1. Generování náhodného výběru a grafické ověřování jeho rozdělení:

1.I. Vygenerujte n = K * 20 náhodných hodnot z rozdělení Exp (L) pomocí inverze distribuční funkce

Příkazy R dle instrukcí:

```
#Reprezentant: Jiří Nádvorník
K = 4
L = 9
n = K*20

#Rovnomerne rozdeleni 'n' hodnot
u = runif(n, min=0, max=1)
u

#transformace dat dle inverzní distribuční funkce. Dle vzorce T= -ln(U)/lambda
#viz
http://en.wikipedia.org/wiki/Exponential_distribution#Generating_exponential_variates
x=-log(1-u)/L
x
```

- Výstup:

```
#vygenerované hodnoty u
[1] 0.54148599 0.85803080 0.72921023 0.52252607 0.11313076 0.38008057 0.13803510
0.38323552
 [9] 0.71394830 0.26228049 0.59609260 0.85200589 0.91957888 0.72501766 0.12704689
0.83084013
[17] 0.80115533 0.95330870 0.21701020 0.96637493 0.77038576 0.18996900 0.27026331
0.16364534
[25] 0.34754161 0.95707879 0.36139388 0.15673130 0.41681142 0.72410153 0.53459173
0.20431153
[33] 0.26960752 0.48756044 0.16450511 0.26967229 0.09051910 0.95996430 0.18190487
0.40235249
[41] 0.23498726 0.94127808 0.71345724 0.11187944 0.19755720 0.68279883 0.75775165
0.97774923
[49] 0.78835727 0.66881043 0.23352326 0.31155224 0.23809001 0.34272596 0.18008339
0.10491599
[57] 0.18949079 0.80483974 0.73005760 0.70599162 0.43874083 0.89848493 0.55724134
0.88002570
[65] 0.06063619 0.75860837 0.72443162 0.63188592 0.55914548 0.44206268 0.27766138
0.17393671
[73] 0.68584253 0.70077001 0.98060062 0.01964657 0.75240413 0.28614286 0.51577530
0.60793441

#vygenerované hodnoty x
[1] 0.086640493 0.216905013 0.145156947 0.082138414 0.013339748 0.053129529 0.016504525
 [8] 0.053696450 0.139064745 0.033799065 0.100729961 0.212286975 0.280053162 0.143449823
[15] 0.015097048 0.197434562 0.179470145 0.340466375 0.027181734 0.376942581 0.163483842
[22] 0.023409196 0.035007945 0.019855836 0.047445324 0.349821014 0.049829714 0.018941070
[29] 0.059916075 0.143080260 0.084982251 0.025394171 0.034908138 0.074285834 0.019970116
[36] 0.034917992 0.010542364 0.357553745 0.022308517 0.057194906 0.029762533 0.314993585
[43] 0.138874166 0.013183087 0.024454967 0.127579901 0.157532428 0.422819843 0.172539519
[50] 0.122784926 0.029550102 0.041479538 0.030214095 0.046628249 0.022061405 0.012315300
[57] 0.023343619 0.181548245 0.145505187 0.136016334 0.064174722 0.254171998 0.090525604
[64] 0.235608632 0.006950270 0.157926071 0.143213276 0.111040266 0.091004482 0.064834295
[71] 0.036140138 0.021231543 0.128651214 0.134060310 0.438057103 0.002204681 0.155106382
[78] 0.037452490 0.080578470 0.104036238
```

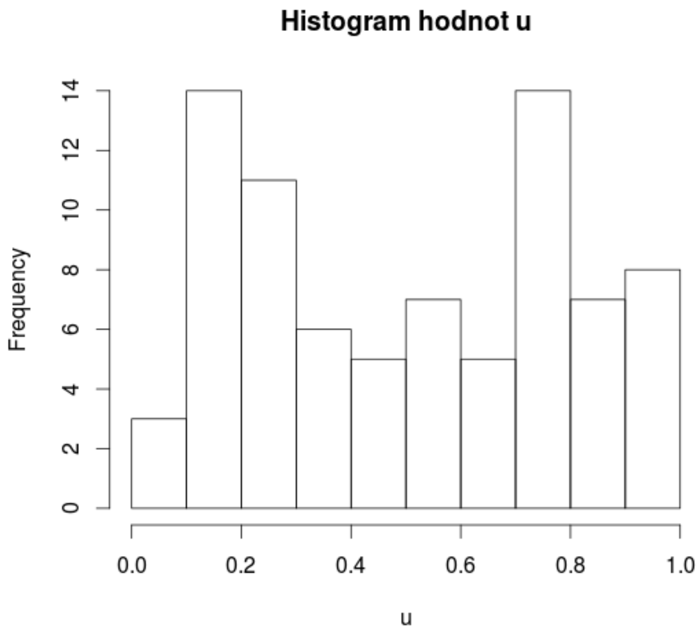
- Proměnná u obsahuje 80 náhodně vygenerovaných hodnoty z uniformního rozdělení, které jsou následně přetransformovány pomocí vzorce x = -ln(U)/L na náhodně vygenerované hodnoty z exponenciálního rozdělení.

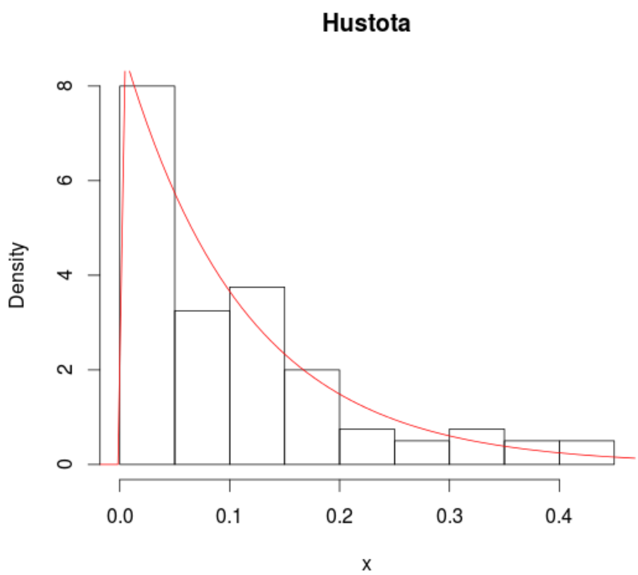
1.II. Vytvořte histogramy dat v u a x. Histogram x zobrazte spolu s grafem hustoty rozdělení Exp (L)

Příkazy R dle instrukcí:

```
hist(u, main="Histogram hodnot u")

#Histogram 'x'  + graf hustoty EXP(L)
hist(x, breaks=3*K, probability=TRUE, main="Hustota")
xWidth=max(x)-min(x)
xGrid=seq(min(x)-0.1*xWidth,max(x)+0.1*xWidth,length=K*20)
lines (xGrid,dexp(xGrid, rate=L), col='red')
```

- Výstup:





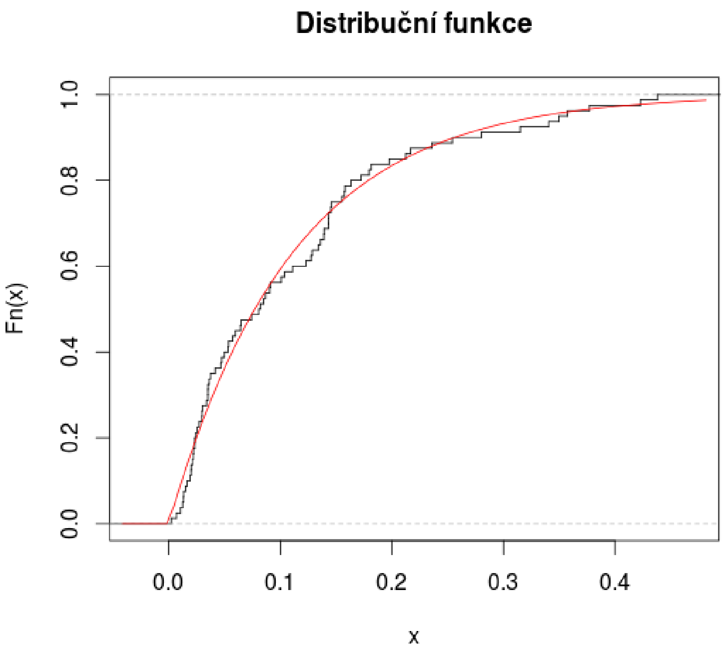
- Na druhém grafu můžeme vidět porovnání hodnot, vygenerovaných pomocí transformace popsané v předchozím úkolu s teoretickým grafem hustoty exponenciálního rozdělení (červeně). Neboť pracujeme s **poměrně malým vzorkem** dat (pouze 80 hodnot), **lze vidět mírné odchylky** ve vygenerovaných hodnotách od teoretické hustoty exponenciálního rozdělení. Nicméně odchylky nejsou nijak extrémní a proto je možné je považovat za postačující vzorek dat.

1.III. Vygenerujte graf empirické distribuční funkce pro data z x spolu s grafem distribuční funkce rozdělení Exp(L)

Příkazy R dle instrukcí:

```
#Empiricka dist. fce 'x' + Distribucni funkce EXP(L)
plot(ecdf(x), verticals=TRUE, do.points = FALSE, main="Distribuční funkce")
lines (xGrid,pexp(xGrid, rate = L), col='red')
```

- Výstup:



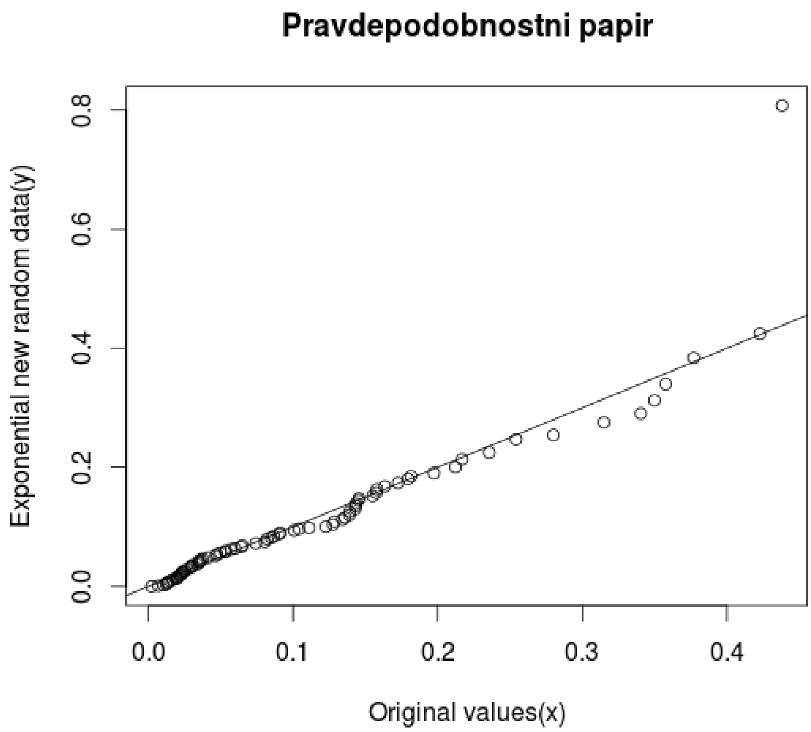
- Graf zobrazuje empirickou distribuční funkci pro data viz výše. Na ose X jsou hodnoty, jakých data nabývají. Na ose Y pak procento, kolik dat ze vzorku těchto hodnot nabývá (přesněji kolik procent dat je dané hodnotně rovno nebo menší). Pří porovnání s teoretickou distribuční funkcí (červenou barvou) je vidno, že ve vygenerovaných datech mají **hodnoty v intervalu 0.1 - 0.15 nižší zastoupení**, zbytek **dobře kopíruje křivku teoretické distr. funkce**.

1.IV. Vygenerujte „pravděpodobnostní papír“ pro porovnání rozdělení dat x s rozdělením Exp(L)

Příkazy R dle instrukcí:

```
#Q-Q plot pro vzorkova data a nove vygenerovana data
y=rexp(1000, rate=L)
qqplot(x,y, plot.it=TRUE, xlab="Original values(x)", ylab="Exponential new random
data(y)", main="Pravdepodobnostni papir")
#referencni osa
abline(a=0, b=1)
```

- Výstup:



- Pravděpodobnostní papír zobrazuje Q-Q Plot našich dat proti nově vygenerovaným datům. Tímto postupem bychom ověřili, že data jsou opravdu z EXP rozdělení. Je vidno, že **data opravdu spadají do stejného rozdělení**, neboť v převážné míře leží na ose. Mírné odbočení od osy v horní části grafu bylo očekáváno, neb tam je již hustota nižší a snadno se tak mohou hodnoty vychylovat. V tomto případě nově vygenerovaný vzorek obsahoval hodnotu 0.81 (bod nejvíce napravo) kdežto nejvyšší hodnota v našich datech byla 0.438.

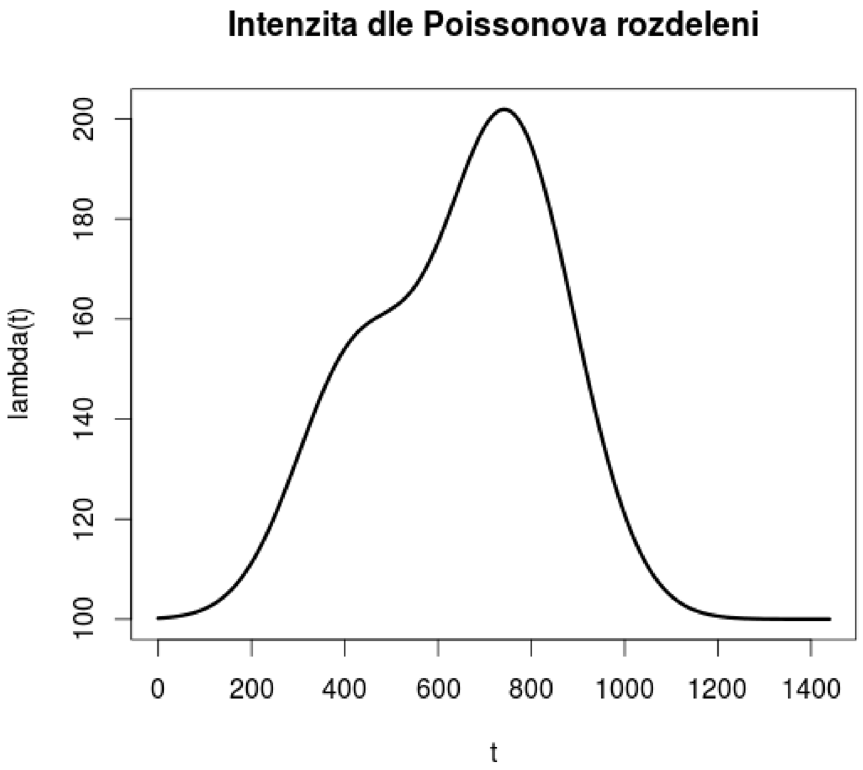
2. Generování nehomogenního Poissonova procesu a grafické ověřování jeho rozdělení:

2.I. Uvažujte příchod požadavků na webový server dle Poissonova procesu s proměnlivou intenzitou lambda(t) příchodů za minutu

Příkazy R dle instrukcí:

```
lambda = function(t) { 100 + 50/exp((t - 420)^2/(3600*L)) + 100/exp((L*(t - 480 -
30*L)^2)/360000)}

#minuty ve 24 hodinach
t=seq(0,24*60-1)
plot (t,lambda(t), lty="solid", lwd=3, type='l', main="Intenzita dle Poissonova
rozdeleni")
```

- Výstup:



- Příchod požadavků na webový server dle Poissonova rozdělení pro zadanou lambdu. Z grafu vyplývá, že nejvíce požadavků chodí na server kolem poledne, kde je peak a následně intenzita požadavků prudce klesá k 100 požadavků/hodinu. Z praxe bychom očekávali, že nejvíce požadavků bude přicházet až tak kolem 2-3 hodiny odpoledne popřípadě kolem 10 hodiny dopoledne. Pro specifické servery však i toto může platit.

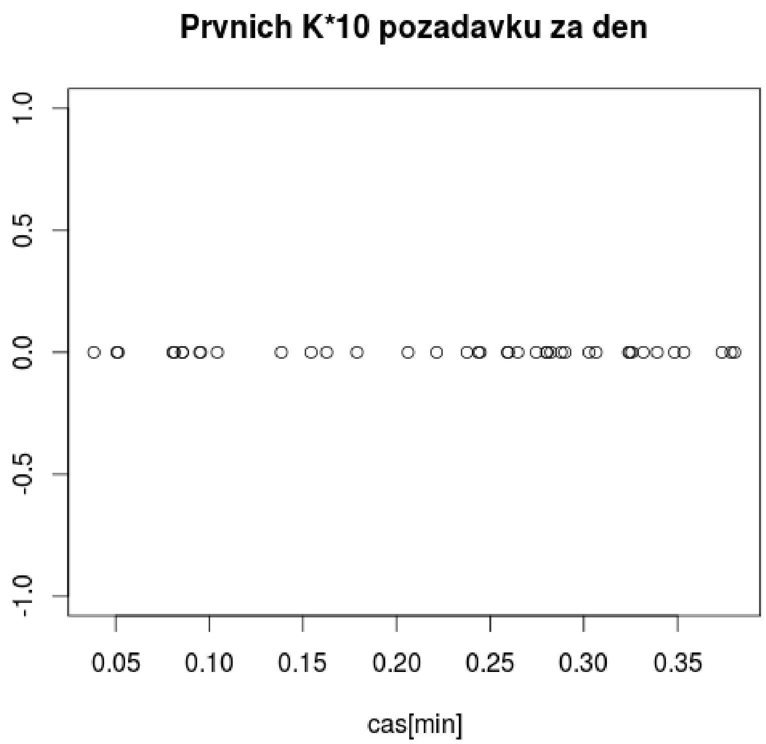
2.II. Vygenerujte časy příchodů požadavků dle tohoto Poissonova procesu pro první den, čili pro první peridu 24*60 minut. Zobrazte prvních K*10příchodů na časové ose.

Příkazy R dle instrukcí:

```
#maximalni hodnota, ktere funkce nabyva
lambda_maximum = optimize(lambda, interval=c(0, 24*60), maximum=TRUE)$objective
t=0
i=0
result = list()

while(t < 24*60){
  lambda_t = lambda(t)

  t = t + rexp(1, lambda_maximum)

  #dalsi vygenerovany cas by byl uz mimo ramec jednoho dne
  #ukoncime cyklus
  if(t > 24*60){
    break
  }

  s = runif(1, min=0, max=1)
  if(s <= lambda_t/lambda_maximum){
          i = i + 1
          result[i] = t
  }
}
```

```
#Vybrani nejvyssi hodnoty ve vysledcich
xWidth=max(unlist(result))
xGrid=seq(min(x)-0.1*xWidth,max(x)+0.1*xWidth,length=K*10)

#Pozadujeme pouze prvnich K*10 vysledku z prvniho dne
result_firstTen = result[0:(K*10)]
plot(result_firstTen,rep(0, length(result_firstTen)), xlab= "cas[min]", ylab = "",
main="Prvnich K*10 pozadavku za den")
```

- Výstup:



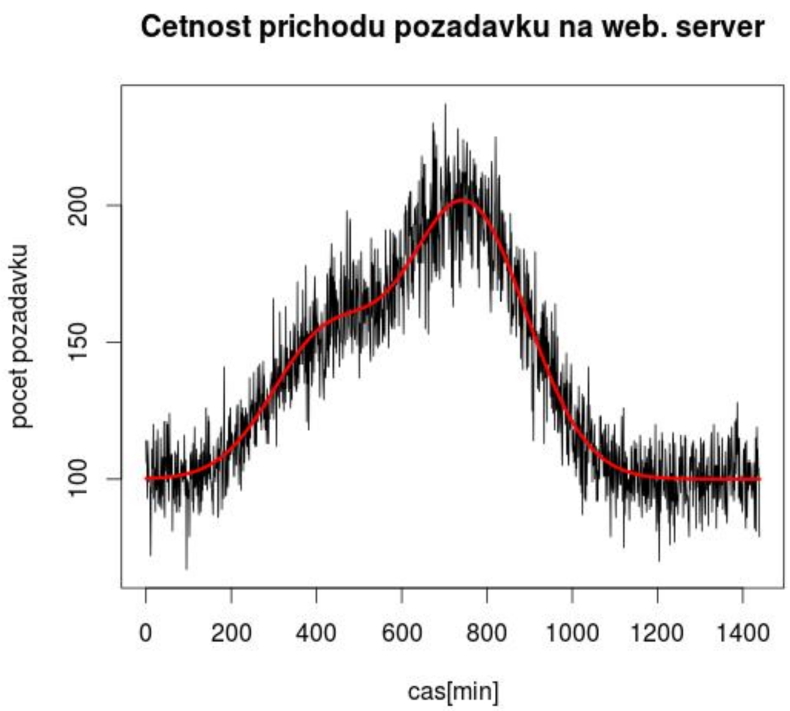
- Na tomto grafu můžeme vidět příchody prvních 40 požadavků na server. Doba pro každý následující výskyt je z exponenciálního rozdělení. Samotné rozložení požadavků na intervalu je tudíž rovnoměrné. Dle grafu intenzity příchodů požadavků z předchozího příkladu lze vyčíst, že **v první minutě by mělo přijít přibližně 100 požadavků** na server. Čili prvních **40 požadavků** by se mělo stihnout přibližně v čase **0.4 minuty**. Pokud se podíváme na výše vygenerovaný graf můžeme říci, že **naše vygenerované hodnoty tomuto předpokladu odpovídají**.

2.III. Zobrazte četnosti příchodů pro celý den, t.j. pro všechy vygenerované příchody během peridy 24*60 minut. Četnosti zobrazte spolu s grafem intenzity lambda(t).

Příkazy R dle instrukcí:

```
t=seq(0,24*60-1)

#Vykresleni krivky za pomoci dat z histogramu
h=hist(unlist(result), breaks=24*60, plot=FALSE)

plot(h$mids, h$counts, type="l", xlab = "cas[min]", ylab="pocet pozadavku",
main="Cetnost prichodu pozadavku na web. server")
lines(t, lambda(t), col="red", lwd=3)
```

- Výstup:



2.IV. Diskutujte kvalitu vygenerovaných dat. Diskutujte efektivitu algoritmu z pohledu ztráty generovaných náhodných hodnot.

- Při zobrazení dat pro celých 24 hodin vidíme, že naše data velice dobře kopírují teoretickou křivku. Vychýlení od křivky dosahují hodnot kolem **25 požadavků/h**, což vzhledem k celkovému objemu požadavků není nijak extrémní. Důležitější je informace, že **hodnoty velice dobře sledují trend teoretické křivky.**

3. Simulace internetového obchodu s tokem nákupů dle nehomogenního Poissonova procesu:

3.I. Uvažujte internetový obchod s tokem objednávek dle nehomogenního Poissonova procesu z předchozího bodu. Ze všech zákazníků K/(K+L)*100% použije kurýrní službu, ostatní zvolí pro doručení objednávky státní poštu. Rozhodnutí o druzích doručení jsou náhodná a nezávislá mezi zákazníky i nezávislá od času zadání objednávky.

3.II. Použijte časů příchodů Poissonova procesu vygenerovaných v předchozím bodě. Rozdělte tento proces na dva procesy:

3.II.a. proces objednávek s použitím kurýrní služby a

3.II.b. proces objednávek s použitím státní pošty.

Příkazy R dle instrukcí:

```
kuryr_prob = K/(K+L)
kuryr_list = list()
posta_list = list()
index=0
while(index <= 24*60){
 lambda_t = lambda(index)
 index_inner=0
 kuryr_list[index] = 0
 posta_list[index] = 0
 while(index_inner < lambda_t){
   r = runif(1, min=0, max=1)
   if(r <= kuryr_prob){
     kuryr_list[index] = as.numeric(kuryr_list[index]) + 1
   }else{
     posta_list[index] = as.numeric(posta_list[index]) + 1
   }
   index_inner = index_inner + 1
 }
 index = index + 1
}
```

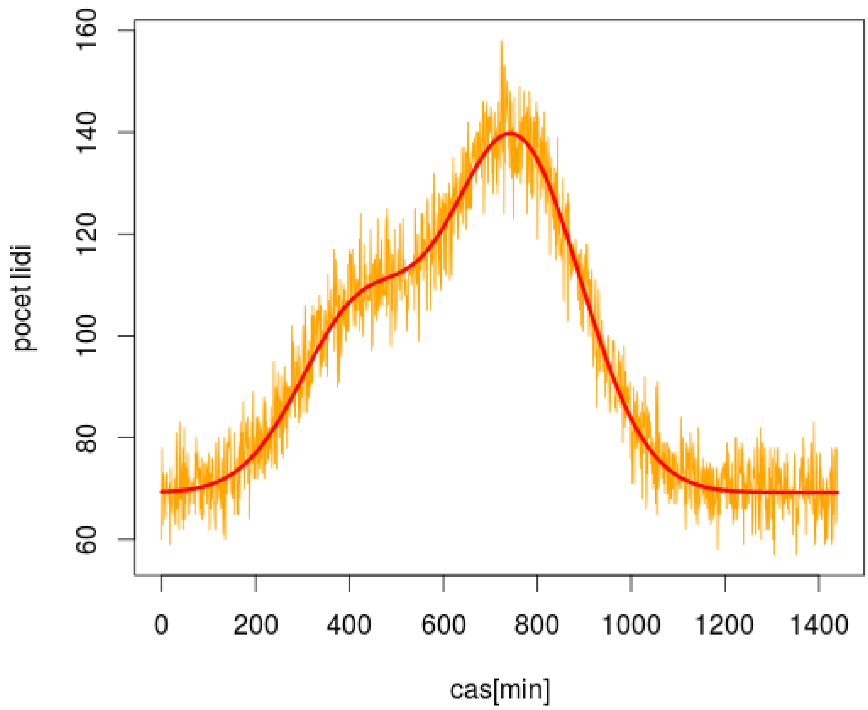
3.III. Zobrazte četnosti příchodů objednávek pro oba procesy pro celý den, t.j. během peridy 24*60 minut. Podobně jako v předchozím bodě, porovnejte četnosti s grafy intenzit lambda_kuryr(t) a lambda_pošta(t).

Příkazy R dle instrukcí:

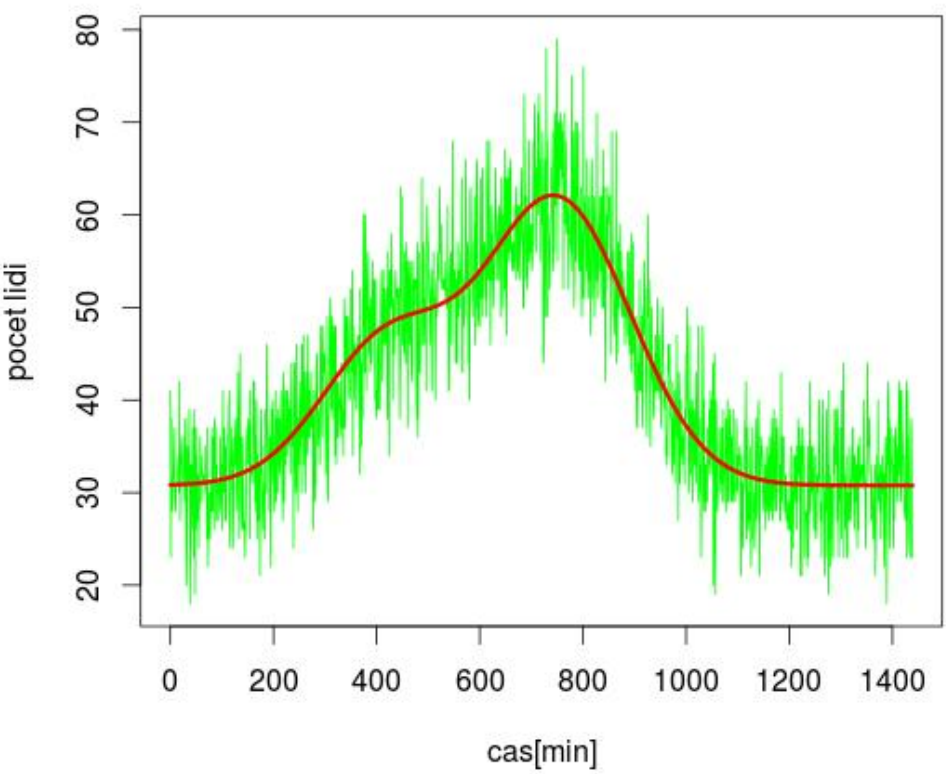
```
#Plot využití pošty
plot(t, posta_list, typ="l", col="orange", xlab = "cas[min]", ylab="pocet lidi",
main="Četnost využití pošty")
lines(t, lambda(t)*(1-kuryr_prob), col="red", lwd="3")

#Plot využití kurýra
plot(t, kuryr_list, typ="l", col="green", xlab = "cas[min]", ylab="pocet lidi", main =
"Četnost využití kurýra")
lines(t, lambda(t)*(kuryr_prob), col="red", lwd="3")


#Plot sloucenych grafu
plot(t, posta_list, typ="l", col="orange", xlab = "cas[min]", ylab="pocet lidi",
main="Porovnání četnosti využití obou služeb"
     ,ylim=c(20,160))
lines(t, lambda(t)*(1-kuryr_prob), col="red", lwd="3")
lines(t, kuryr_list, typ = "l", col="green")
lines(t, lambda(t)*(kuryr_prob), col="red", lwd="3")
```

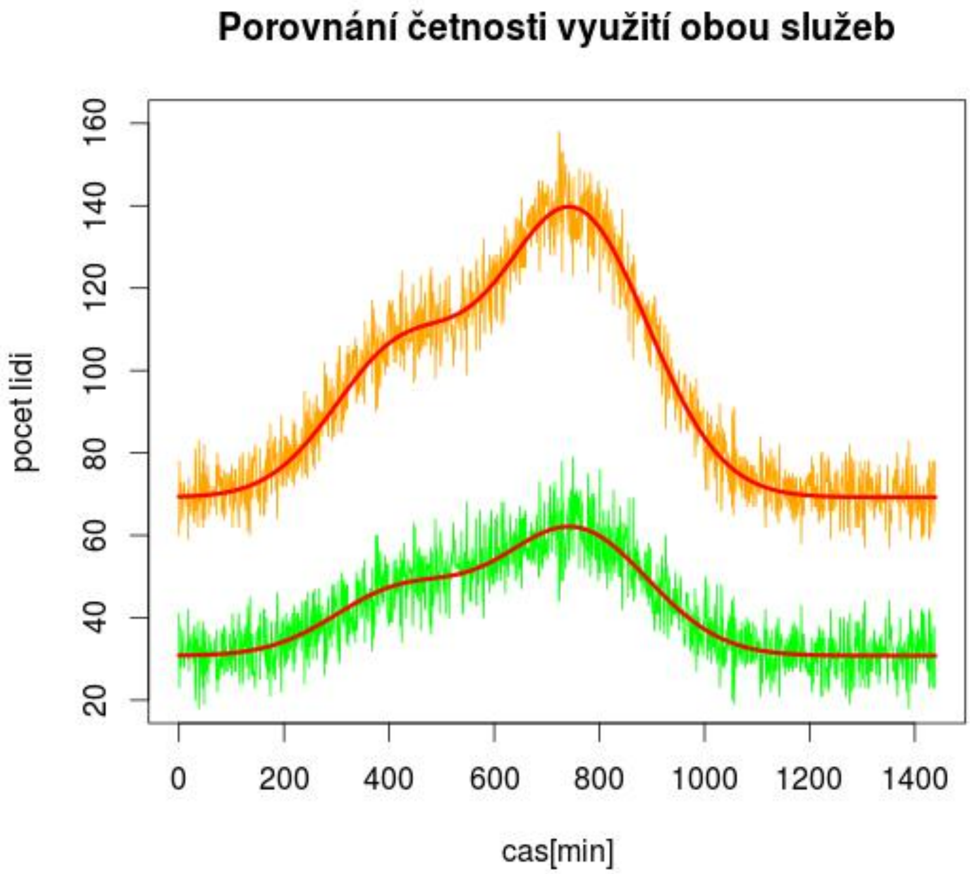
- Výstup:

Četnost využití pošty



Četnost využití kurýra





- V porovnání obou grafů je zřetelně vidět, že četnost využití pošty je přibližně **70%** a kurýra zbylých **30%**. To v prvních minutách odpovídá zhruba 30 kurýrům a 70 poštám neboť celkově je přibližně 100 zakázníků. Dále je vidět, že **oba grafy sledují stejný trend**, který je společný pro všechny grafy odvozené od intenzity dle Poissonova rozdělení. Sečtením těchto dvou grafů bych pak následně získal graf zobrazený v předchozí úloze.