

MI-SPI 2015 - Domácí úkol č.1

Vedoucí týmu: Tomáš Nesrovnal (*nesrotom, 107*)

Členové týmu: Vojtěch Krákora (*krakovoj, 103*), Tomáš Šabata (*sabattom, 103*)

Datum: 22.4.2015

```
1 K = nchar('Tomas') # jmeno
2 L = nchar('Nesrovnal') # prijmeni
```

1 Generování náhodného výběru a grafické ověřování jeho rozdělení:

1.1 Generování náhodného výběru

```
1 n = K*20
2 u=runif(n, min=0, max=1) # Generuje n náhodných hodnot z UNIF(0,1)
3 x=-log(1-u)/L # Exp rozdělení z rovnomerného inverzní distr. fcí.
```

1.2 Vytvoření histogramu

1.2.1 Histogram rovnně rozdělených náhodných hodnot

```
1 hist(u, freq=FALSE)
```

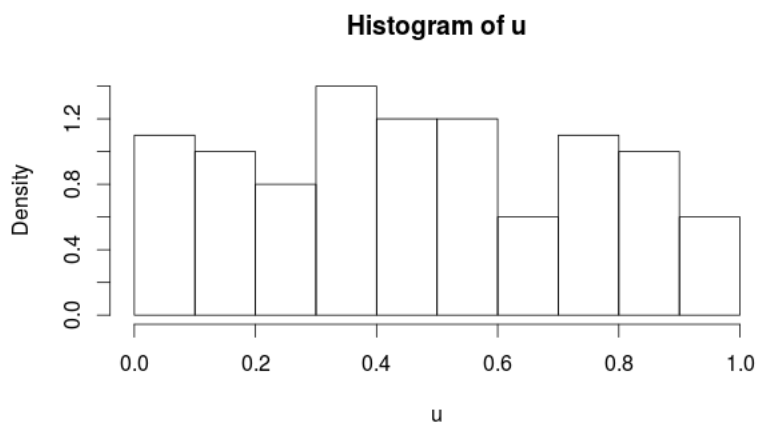


Figure 1: Histogram náhodného rozdělení.

1.2.2 Histogram exponenciálního rozdělení

```
1 hist(x, breaks=5*K, freq=FALSE)
2 xGrid=seq(min(x)-0.2*xWidth, max(x)+0.2*xWidth, length=n)
3 lines (xGrid, dexp(xGrid, rate=L), col='red')
```

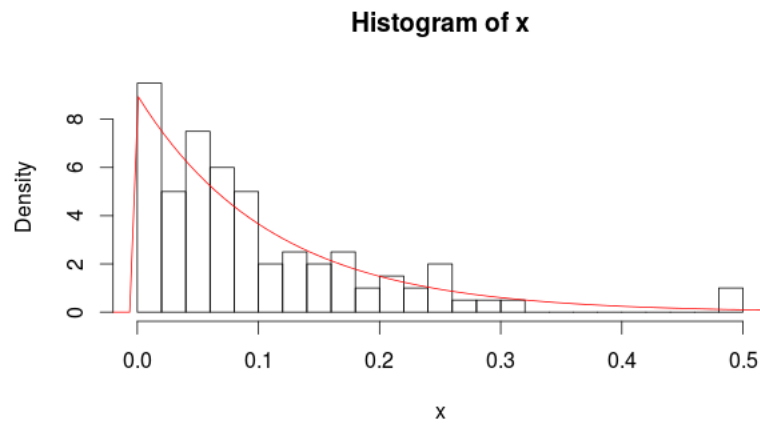


Figure 2: Histogram náhodného rozdělení.

1.3 Graf empirické distribuční funkce

```
1 plot(ecdf(x), verticals=TRUE, do.points = FALSE)
2 lines (xGrid, pexp(xGrid, rate = L), col='red')
```

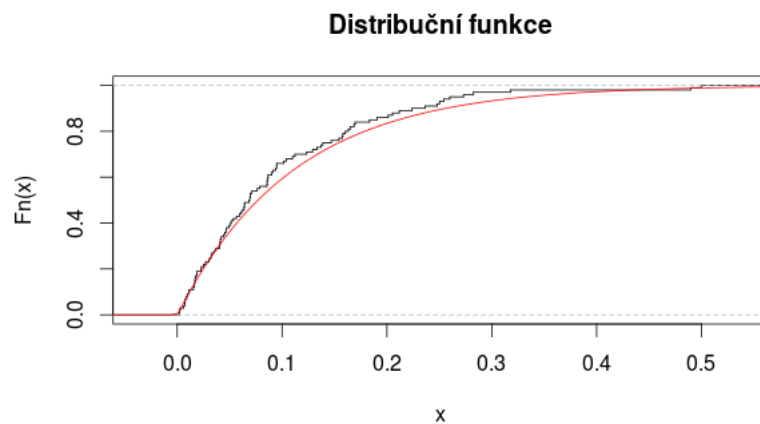


Figure 3: Graf empirické distribuční funkce spolu s grafem $\text{Exp}(L)$

1.4 Generování pravděpodobnostního papíru

```
1 y=rexp(n, rate=L)
2 qqplot(x, y)
3 abline(0,1, col='red', lwd=2)
```

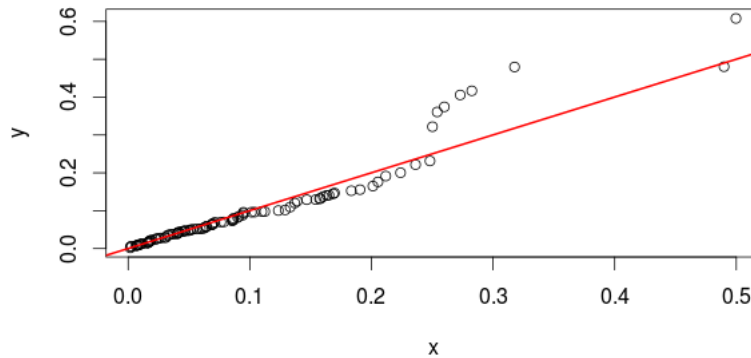


Figure 4: Porovnání dat x s rozdělením $\text{Exp}(L)$

1.5 Diskuze kvality vygenerovaných dat

1.6 Test dobré shody

Vstup:

```
1 chisq.test(x, p=y/sum(y))
```

Výstup

```
1 data: x
2 X-squared = 57.6125, df = 99, p-value = 0.9997
```

2 Generování nehomogenního Poissonova procesu

2.1 Intenzita příchodů požadavků na server

```
1 lambda = function(t){100+50*exp(-(t-420)^2/(3600*L))+100*
2                      exp(-(L*(-30*L+t-480)^2)/360000)}
3 # první perioda
4 t=seq(0,3*(24*60)-1)
```

```

5 plot(t, lambda(t%%(24*60)), lty="solid", lwd=3, type='l',
6      main="Intenzita řůpřístup za den",
7      ylab="řPřichody za minutu",
8      xlab="řas t v minutách")
9 text(0, 200, "Den 1", cex=0.6, pos=4)
10 text(24*60, 200, "Den 2", cex=0.6, pos=4)
11 text(24*120, 200, "Den 3", cex=0.6, pos=4)
12 abline(h = 0, v = 01)
13 abline(h = 0, v = 24*60)
14 abline(h = 0, v = 24*120)

```

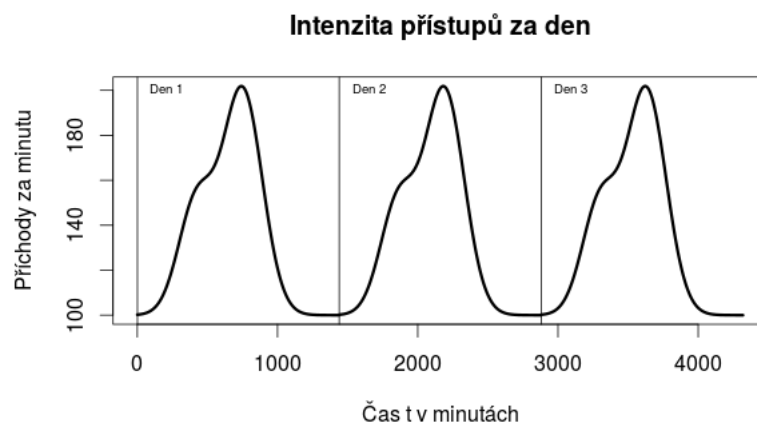


Figure 5: Intenzita přřichodů požadavků na server za řři dny

2.2 Generování řasů přřichodů

```

1 p = K*10 # přřichod
2 event = numeric(p)
3 s = 10^-(2.4) # step
4 t = 0 # time = Aktuální řas
5 i = p # iterator
6
7 # Cyklus simuluje ubřřhající řas (po skocích delta) a na řáklade
8 # funkce lambda generuje řpřichod řákazníka
9 while (0 < i) {
10   if (runif(1, min=0, max=1) < lambda(t) * s) {
11     i = i - 1
12     event[i] = t # oznacime řas udalosti
13   }
14   t = t + s
15 }
16 plot(event, numeric(p))

```

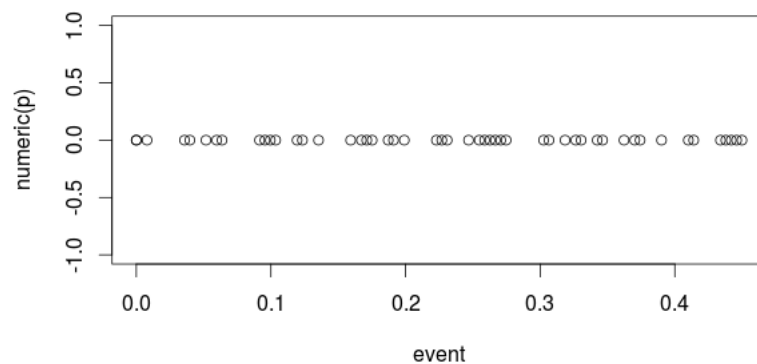


Figure 6: Vygenerované časy příchodů.

2.3 Zobrazení četnosti příchodů

1 Zatím nic

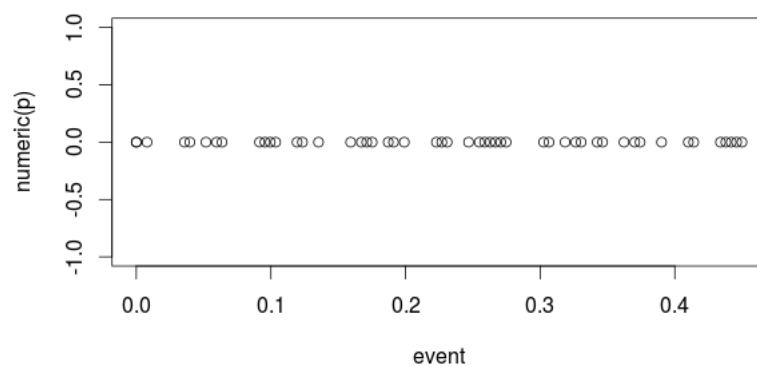


Figure 7: Vygenerované časy příchodů.

2.4 Diskuze kvality vygenerovaných dat