cvic10 – bodové odhady

Bodový odhad pro $U(0, \vartheta)$

```
theta = 4.7    # parametr, který ve skutečné aplikaci NEZNÁME
n = 10         # počet měření, tzv. rozsah náhodného výběru

t = runif(n,0,theta)  # realizace náhodného výběru ... tj. konkrétní n-tice čísel
t
```

```
##  [1] 1.8797472 3.1566634 1.7103418 2.1326453 3.2704093 1.3571119 0.5036919
##  [8] 0.8267204 1.4666162 0.8729825
```

Jeden možný bodový odhad: dvojnásobek průměru z čísel, co jsme dostali. A jeho kvadratická chyba.

```
theta2 = 2*mean(t)
theta2
```

```
## [1] 3.435386
```

```
(theta2-theta)^2
```

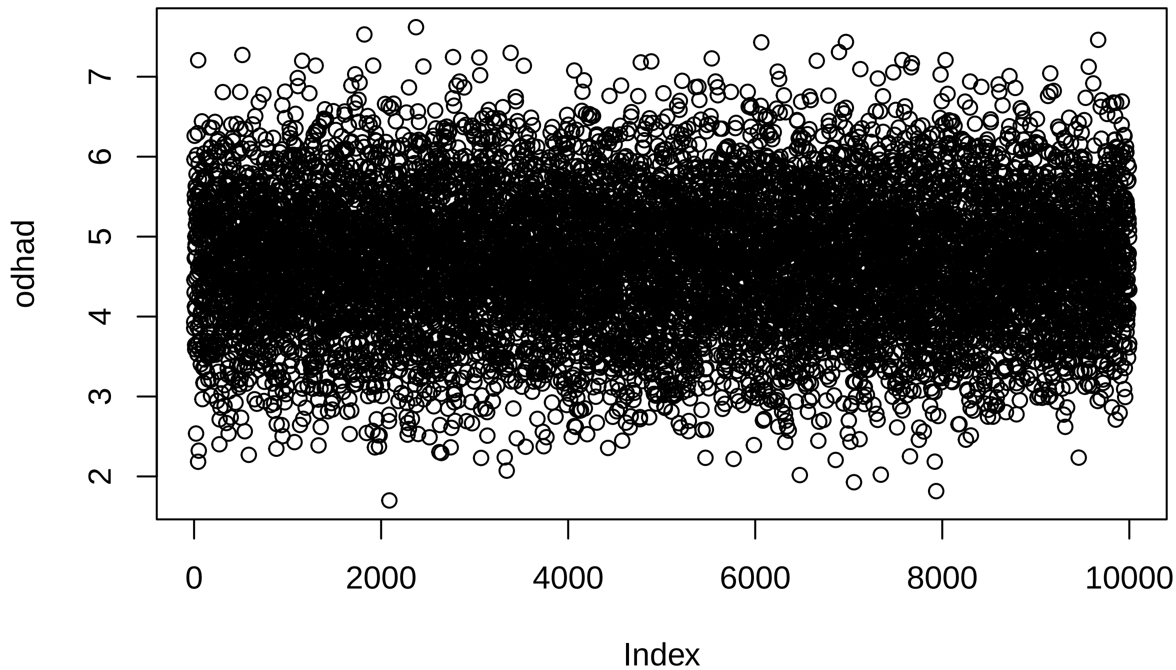
```
## [1] 1.599249
```

Co znamenají vlastnosti bodového odhadu? A jak je ověřit? Např. nestrannost: jde nám o střední hodnotu výrazu theta2 = 2*mean(t), chceme, aby se rovnala theta. To můžeme spočíst “teoreticky na papíře” (a to přesně), tady zkusme samplovat. Budeme tedy opakovat celý pokus znovu a znovu a dělat průměr výrazu 2*mean(x). Pro přehlednost for-cyklem, abychom neodváděli pozornost k elegantním R-kovým konstrukcím.

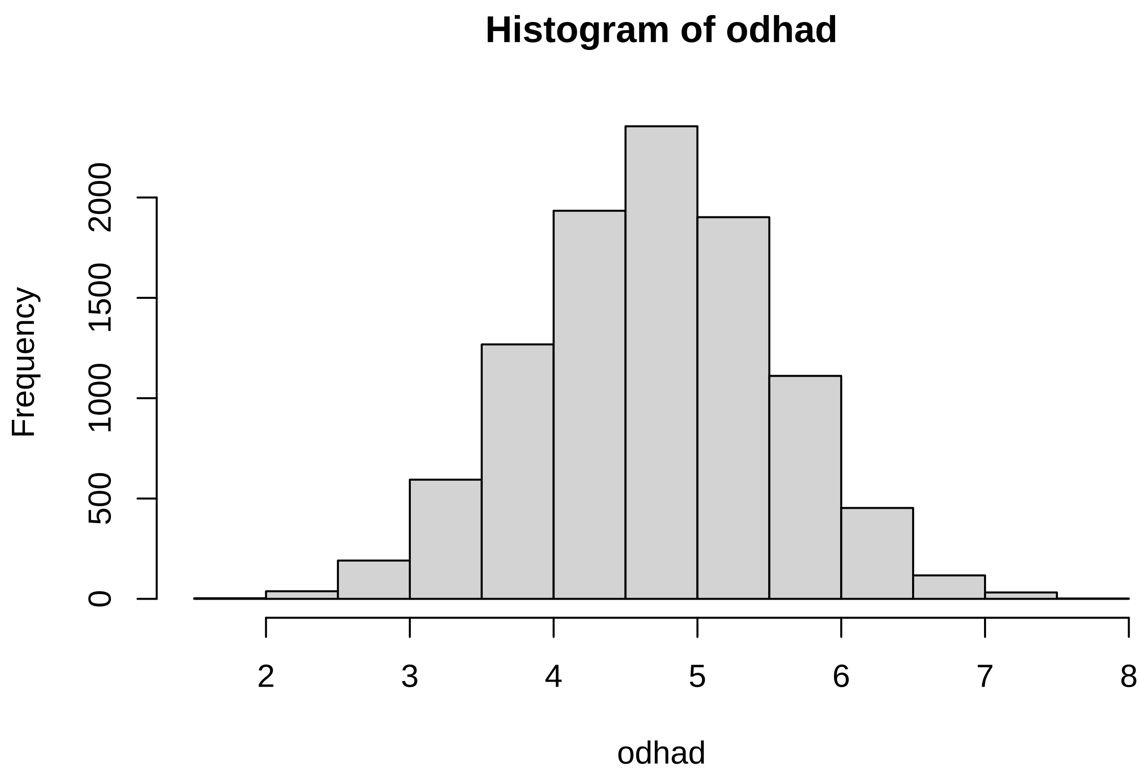
```
N = 10^4
odhad = rep(0,N)

for (cnt in 1:N) {
  odhad[cnt] = 2*mean(runif(n,0,theta))
}

plot(odhad, type='p')
```




```
hist(odhad)
```




```
mean(odhad)
```

```
## [1] 4.69458
```

```
mean((odhad-theta)^2)
```

```
## [1] 0.7164085
```

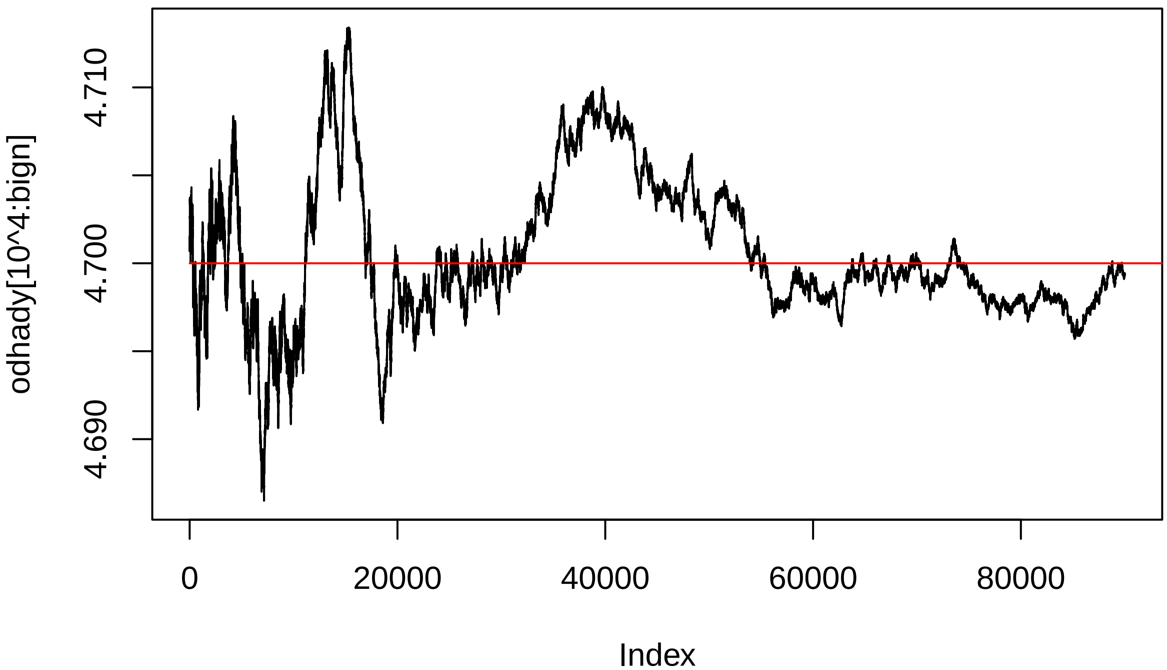
Konzistence znamená, že pro velká n dostaneme "skoro vždy" správnou hodnotu.

```
bign = 10^5
bigx = runif(bign,0,theta)

#hloupý způsob jak získat průměry ze všech prefixů: v kvadratickém čase
#odhady = 0*bigx
#for (n in 1:bign) {
#  odhady[n] = 2*mean(bigx[1:n])
#}

#chytřejší způsob v lineárním čase:
odhady = 2*cumsum(bigx)/seq(1:bign)

plot(odhady[10^4:bign], pch=2, type='l', cex=1)
lines(c(0,bign),c(4.7,4.7),col='red')
```



Teď totéž pro odhad získaný metodou max. věrohodnosti.

```
theta = 4.7    # parametr, který ve skutečné aplikaci NEZNÁME
n = 10         # počet měření, tzv. rozsah náhodného výběru

t = runif(n,0,theta)  # realizace náhodného výběru ... tj. konkrétní n-tice čísel
t
```

```
##  [1] 2.07322054 3.27816331 0.98977171 2.15294129 4.37184624 3.33640660
##  [7] 1.44981397 0.01378183 1.97685402 0.65368135
```

```
max(t)  #*(n+1)/n
```

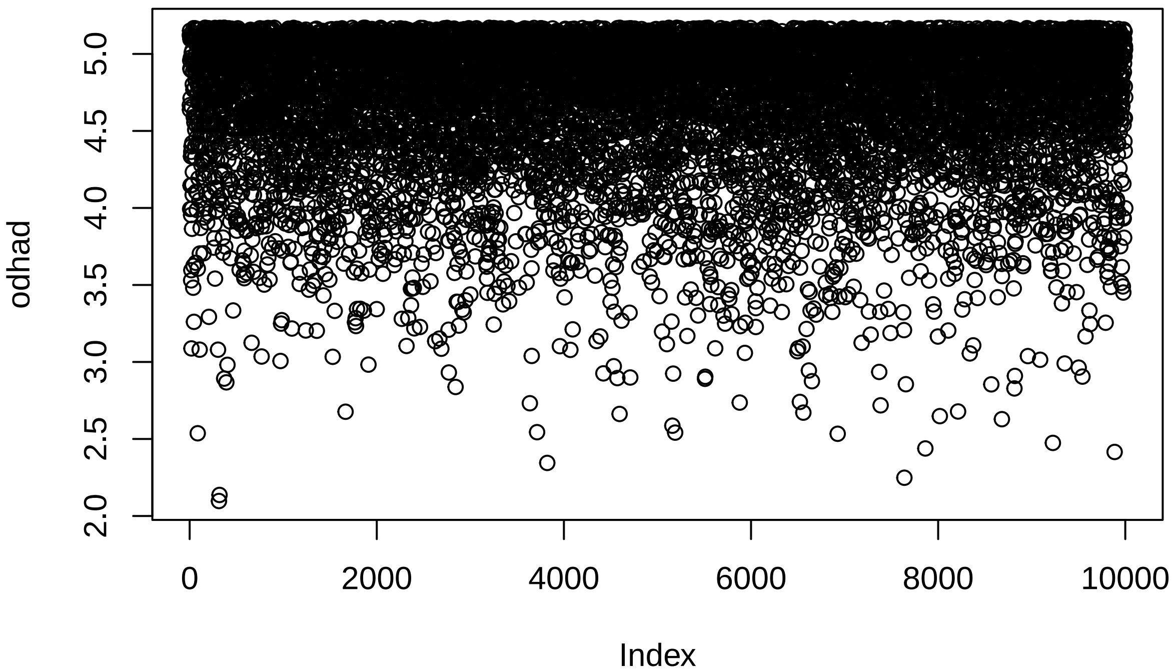
```
## [1] 4.371846
```

Nestrannost a MSE.

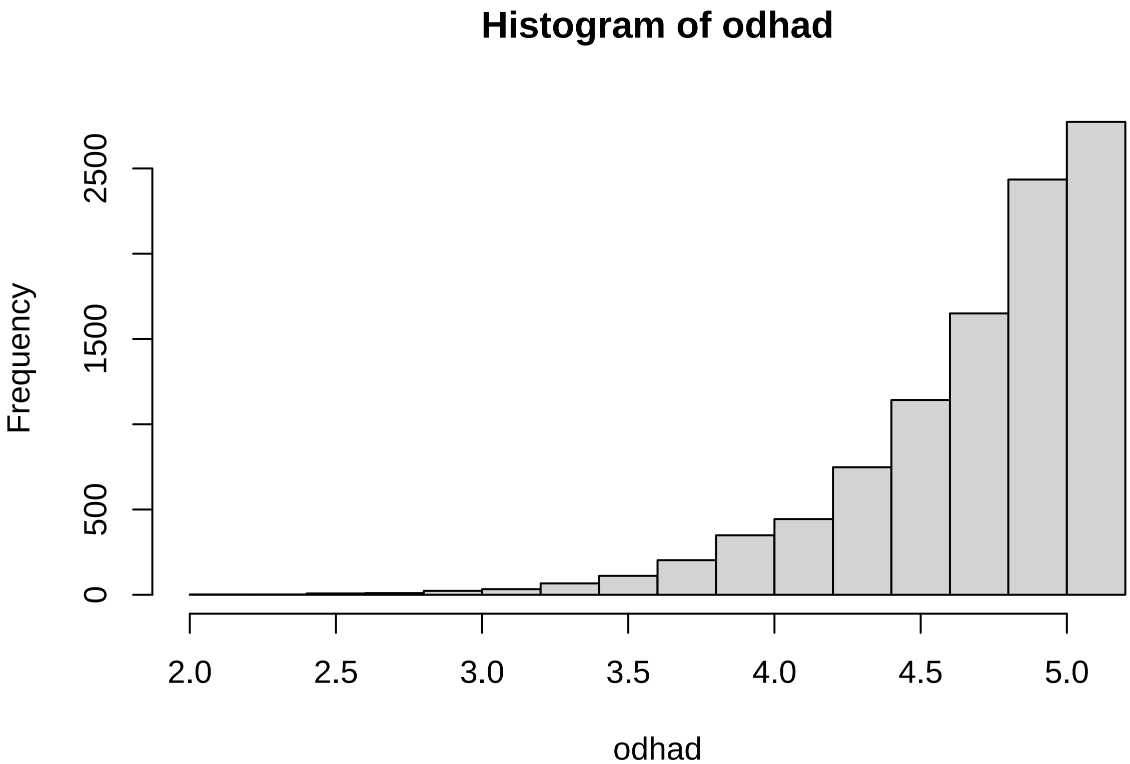
```
N = 10^4
odhad = rep(0,N)
```

```r
for (cnt in 1:N) {
  odhad[cnt] = max(runif(n,0,theta))*(n+1)/n
}

plot(odhad)
```




```r
hist(odhad)
```



```r
mean(odhad)
```

```
## [1] 4.696983
```

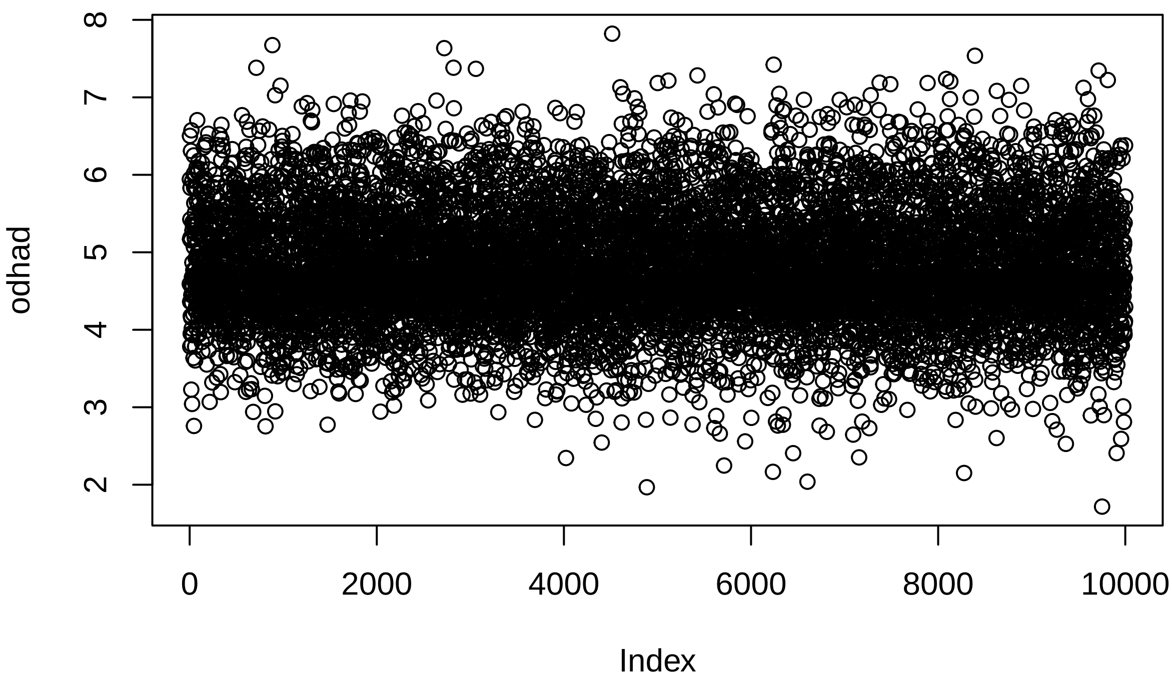
```r
mean((odhad-theta)^2)
```

```
## [1] 0.1838378
```

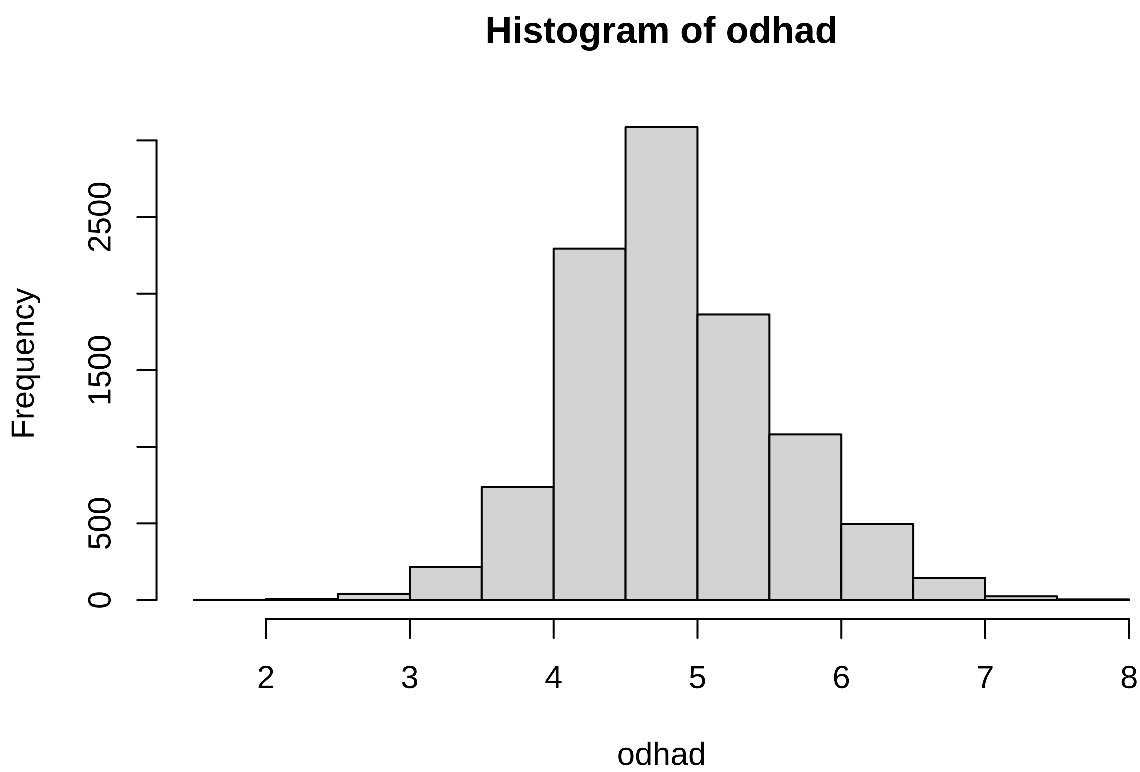
```r
N = 10^4
odhad = rep(0,N)

for (cnt in 1:N) {
  t = runif(n,0,theta)
  a = max(t) #*(n+1)/n
  b = 2*mean(t)
  odhad[cnt] = max(a,b)
}

plot(odhad)
```




```r
hist(odhad)
```




```
mean(odhad)
```

```
## [1] 4.824534
```

```
mean((odhad-theta)^2)
```

```
## [1] 0.5354555
```

```
MSE_point <- function(m=0.5, rep=10^3, n=10, DATA=runif, EST=mean) {
  mean(replicate(rep, (EST(DATA(n))-m)^2))
}
```

```
twomean <- function(data) {
  2*mean(data)
}

twomedian <- function(data) {
  2*median(data)
}

mmax <- function(data) {
  n = length(data)
  (n+1)/n*max(data)
}

#MSE_point(m=0.5, rep=10^4, DATA=runif, EST=mean)
#MSE_point(m=0.5, rep=10^4, DATA=runif, EST=median)

#MSE_point(m=0, rep=10^4, DATA=rnorm, EST=mean)
#MSE_point(m=0, rep=10^4, DATA=rnorm, EST=median)

MSE_point(m=1, rep=10^4, DATA=runif, EST=twomedian)
```

```
## [1] 0.07702458
```

```r
MSE_point(m=1, rep=10^4, DATA=runif, EST=twomean)
```

```
## [1] 0.03380679
```

```r
MSE_point(m=1, rep=10^4, DATA=runif, EST=max)
```

```
## [1] 0.01487223
```

```r
MSE_point(m=1, rep=10^4, DATA=runif, EST=mmax)
```

```
## [1] 0.008409304
```