

cvic13

Příklad 1 – test

1(a)

Zkusíme nasamplovat – jen jedna otázka se třemi volbami.

```
n = 10^4
p = 0.5

umi = rbinom(n, 1, p) # umi je vektor 0/1
#umi
spravne = umi + (1-umi)*rbinom(n,1,1/3) # taky 0/1 (jestli je odp. spravne)
#spravne
#umi*spravne == umi
sum(umi*spravne)/sum(spravne) # sample podmíněné pravděpodobnosti

## [1] 0.7483774
```

1(c)

Celý test s deseti otázkami. Nasamplujeme distribuci otázek co umí, pokud odpověděl správně 6. Nakreslíme do grafu spolu se spočteným vzorcem.

```
n = 10^4
p = 0.5

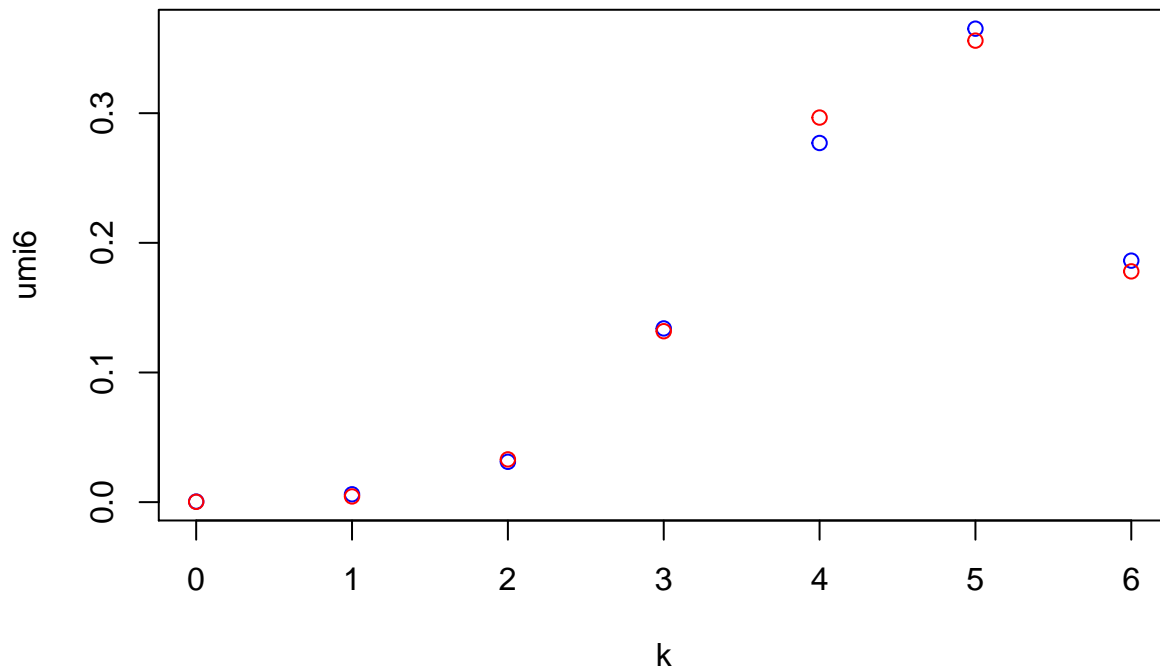
# nejprve jednotlivé otázky, jako v 1(a)
umi = rbinom(10*n, 1, p)
spravne = umi + (1-umi)*rbinom(10*n,1,1/3)

# teď posčítáme jednotlivé desítky
celk_spravne = tapply(spravne, (seq_along(spravne)-1) %/% 10, sum)
celk_umi = tapply(umi, (seq_along(umi)-1) %/% 10, sum)

# a teď vytáhneme, kolik otázek uměl student, co odpověděl správně 6 otázek
k = 0:6
umi6 = rep(0,7)

for (i in k) {
  # print(i)
  umi6[i+1] = (sum(celk_umi == i & celk_spravne == 6)/sum(celk_spravne == 6))
}

plot(k,umi6, col='blue')
points(k, dbinom(k,6,3/4), col='red')
```

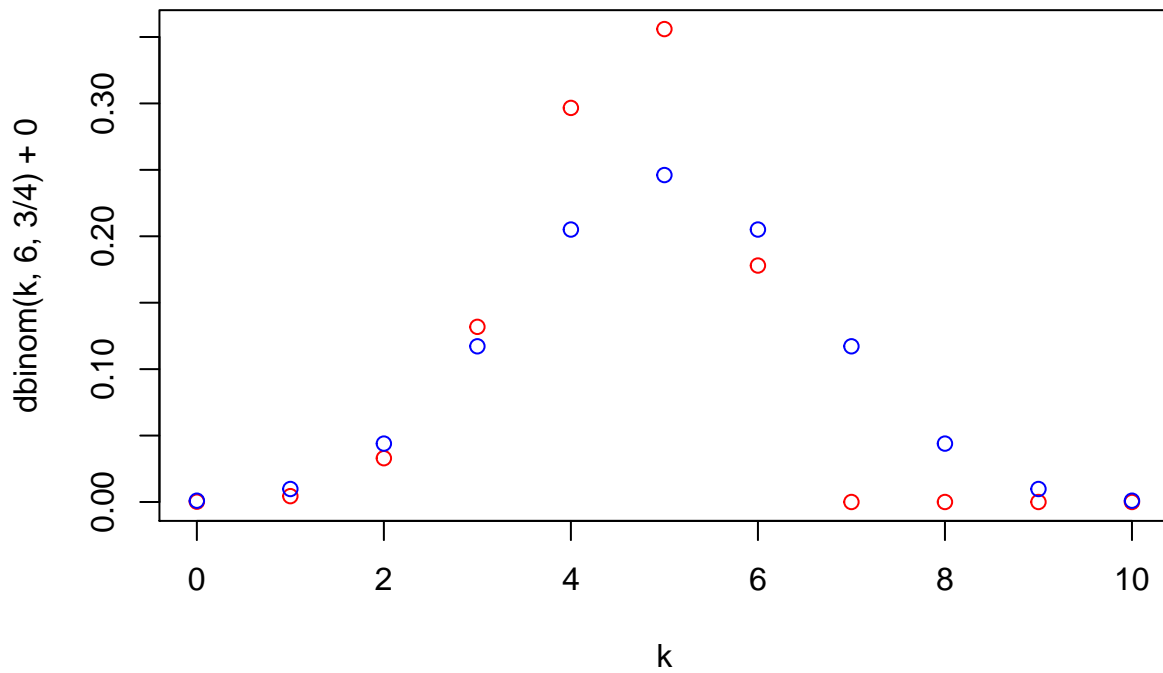


1(b,c) – Rozdíl apriori & aposteriori.

I když se to nezdá, tak 6 správných je podprůměrná odpověď (jaká je střední hodnota počtu správných odpovědí?), proto se náš odhad, kolik student uměl, snížil oproti stavu před testem.

`k=0:10`

```
plot(k, dbinom(k,6,3/4)+.0, col='red')
points(k, dbinom(k,10,1/2), col='blue')
```



Příklad 2 – hodně testů ## (2a)

Použijeme MAP pravidlo pro rozhodnutí.

```

pp <- function(p) { p + (1-p)/3 }

k = 0:10
pp(0.3)

## [1] 0.5333333

pp(0.7)

## [1] 0.8

A = dbinom(k,10,pp(0.3)); A

## [1] 0.0004898553 0.0055983463 0.0287914951 0.0877455088 0.1754910176
## [6] 0.2406733955 0.2292127577 0.1496899642 0.0641528418 0.0162927852
## [11] 0.0018620326

B = dbinom(k,10,pp(0.7)); B

## [1] 0.0000001024 0.0000040960 0.0000737280 0.0007864320 0.0055050240
## [6] 0.0264241152 0.0880803840 0.2013265920 0.3019898880 0.2684354560
## [11] 0.1073741824

C = dbinom(k,10,pp(0.95)); C

## [1] 1.693509e-15 4.911175e-13 6.409084e-11 4.956358e-09 2.515352e-07
## [6] 8.753424e-06 2.115411e-04 3.505538e-03 3.812273e-02 2.456798e-01
## [11] 7.124714e-01

AA = A/(A+B+C) # posteriorní pravd. v závislosti na k
BB = B/(A+B+C)
CC = C/(A+B+C)
print("")

## [1] ""

AA

## [1] 0.999791002 0.999268890 0.997445782 0.991116913 0.969583490 0.901039877
## [7] 0.721919298 0.422230283 0.158689893 0.030717456 0.002266053

BB

## [1] 0.0002089976 0.0007311097 0.0025542155 0.0088830308 0.0304151201
## [6] 0.0989273512 0.2774144408 0.5678816505 0.7470088862 0.5060923661
## [11] 0.1306720070

CC

## [1] 3.456439e-12 8.766133e-11 2.220348e-09 5.598384e-08 1.389726e-06
## [6] 3.277132e-05 6.662613e-04 9.888066e-03 9.430122e-02 4.631902e-01
## [11] 8.670619e-01

#A (typ 0.3) je nejpravděpodobnější pro hodnoty:
k[A>B & A>C]

## [1] 0 1 2 3 4 5 6

k[AA>BB & AA>CC]

## [1] 0 1 2 3 4 5 6

```

```
#B (typ 0.7) je nejpravděpodobnější pro hodnoty:  
k[B>A & B>C]
```

```
## [1] 7 8 9
```

```
k[BB>AA & BB>CC]
```

```
## [1] 7 8 9
```

```
#C (typ 0.95) je nejpravděpodobnější pro hodnoty:  
k[C>A & C>B]
```

```
## [1] 10
```

```
k[CC>AA & CC>BB]
```

```
## [1] 10
```

(2b)

MAP pravidlo – najdeme max. ve vektoru pu

podm. stř. – spočítáme podm. střední hodnotu

```
l = 0:5
```

```
db = function(p) { dbinom(1,5,p/(1/3+(2/3)*p)) }
```

```
pu = (db(0.3)+db(0.7)+db(0.95))/3
```

```
pu; sum(pu)
```

```
## [1] 0.005352974 0.034702764 0.093320816 0.149386275 0.221924706 0.495312464
```

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

```
#MAP pravidlo:  
l[pu == max(pu)]
```

```
## [1] 5
```

```
#podm. střední hodnota  
sum(pu*l)
```

```
## [1] 4.033764
```

Příklad 3

Zkusíme nasamplovat pravidlo: pokud vytáhnú bílý míček, odpovím první krabička. Kódování míčku: 1 = bílá, 2 = černá Kódování krabiček: 1, 2 Pro $p > 1/3$ je MAP-pravidlo pro náš tip na krabičku barva míčku.

```
n = 10^4  
chyb = 0  
p = 0.5  
for (i in 1:n){  
  krab = ifelse(runif(1)<p,1,2)  
  if (krab==1) {  
    micek = sample(c(1,1,2),1)  
  } else {  
    micek = sample(c(1,2,2),1)  
  }  
}
```

```

}

if (micek != krab) {
  chyb = chyb + 1
}
}
print(chyb/n)

```

```
## [1] 0.3362
```

Příklad 4

```

theta = seq(0,1,.005)
k=4; n=10

plot(theta,
      theta^(k+1)*(1-theta)^(n-k)*(theta<=0.5)+
      theta^(k)*(1-theta)^(n+1-k)*(theta>0.5),
      type='l',
      lty = 1, lwd = 2
    )

```

