Testy o střední hodnotě při neznámém rozptylu

Mgr. Zdeňka Geršlová

Teorie

Studijní materiály k přednáškám: stka_STATINF_statinference_handout_02_2024_draft.pdf

kniha Aplikovaná štatistická inferencia I.,
kapitola 6.1 (od str. 143)

Praktický příklad

Z archivních materiálů (Schmidt, 1888) máme k dispozici původní kraniometrické údaje 215 dospělých mužů a 107 dospělých žen ze starověké egyptské populace o basion–bregmatické výšce lebky. Údaje jsou k dispozici v souboru `two-samples-means-skull.txt`. Současně máme k dispozici hodnoty basion–bregmatické výšky ($\bar{x}_m = 133.977$,mm; $\bar{x}_f = 126.942$,mm) a hodnoty směrodatné odchylky ($s_m = 5.171$,mm; $s_f = 4.430$,mm) mužů a žen novověké egyptské populace.

Na hladině významnosti $\alpha = 0.05$ otestujte hypotézu o shodě střední hodnoty basion-bregmatické výšky lebky žen staroegyptské populace se střední hodnotou basion-bregmatické výšky lebky žen novověké egyptské populace, tj. nulovou hypotézu
$H_0 : \mu_f = 126.942$ oproti alternativní hypotéze
$H_1 : \mu_f \neq 126.942$.
Před testováním ověřte předpoklad normality.
Testování proveďte pomocí
(a) kritického oboru;
(b) intervalu spolehlivosti;
(c) p-hodnoty
při použití testovacích statistik (1) T_W, (2) U_W, (3) U_S, (4) $U_L R$.

Dále vykreslete grafy zobrazující věrohodnostní intervaly spolehlivosti pro test o střední hodnotě získané na základě testovacích statistik U_W, U_S a U_{LR}.

Načtení dat

Načteme data, vyfiltrujeme (pomocí `==`) pouze ženy a proměnnou `skull.H`, odstraníme NA hodnoty. Střední hodnotu a rozptyl neznáme, proto je odhadneme pomocí `mean()` a `var()`, resp. `sd()`.

```
read.delim("cesta k souboru", sep = '\t', dec = '.')
skull.F <- data[data$sex == 'f', 'skull.H']
skull.F <- na.omit(skull.F)
```

$\hat{\mu} = 125.682243$

$\hat{s} = 4.6532564$

Ověření normality

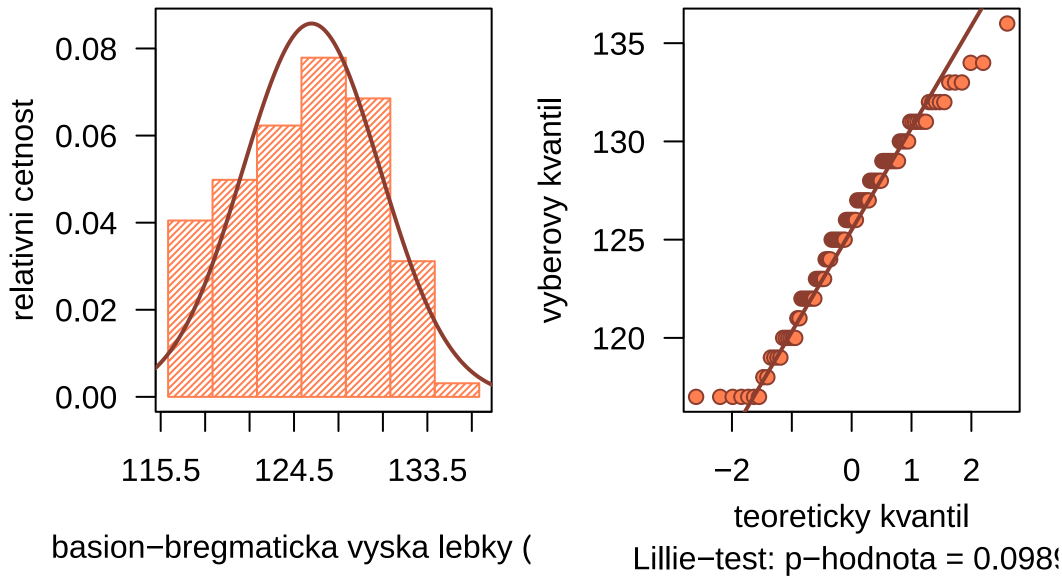
Normalitu ověříme graficky (histogram, qq-plot) a testem (adekvátně rozsahu náh. výběru). U histogramu je nutné nastavit počet třídicích intervalů pomocí Sturgesova pravidla

```
r <- 3.3 + log10(n) + 1 # 6.32 -> 7
range(skull.F) # 117 136 -> 19 -> 21/7 = 3
```

Zaokrouhlíme na nejbližší vyšší celé číslo, spočítáme rozsah proměnné a adekvátně ho rozšíříme tak, aby byl dělitelný tímto číslem, čímž získáme šířku třídicích intervalů. Tedy v našem případě rozšíříme rozsah na 116 - 137 a délka intervalu je 3.

```
b <- seq(116, 137, by = 3)
hist(..., breaks = b, ...)
lines(..., dnorm(..., mean(skull.F), sd(skull.F)), ...)

qqnorm(skull.F, ylab = 'vyberovy kvantil', ...)
qqline(skull.F, ...)
mtext('teoreticky kvantil', side = 1, line = 2.2)
mtext(paste('Lillie-test: p-hodnota =', round(nortest::lillie.test(skull.F)$p.val, 4)), si
# zpusob, jak zadat vysledek testu do popisku grafu
```

Grafy pro ověření normality



Protože p-hodnota Lillieforsova testu je ………, zamítáme/nezamítáme H0 na hladině významnosti 0.05, a tedy předpokládáme, že data pocházejí/nepocházejí z normálního rozdělení. Vizuálním posouzením vidíme/nevidíme výrazné odchylky od normality.

Testovací statistiky

Vypočítáme testovací statistiku t_W a ostatní statistiky vypočítáme pomocí vztahů mezi jednotlivými typy statistik:

$u_W = \frac{n}{n-1} t_W^2$, $u_S = \frac{n t_W^2}{n-1+t_W^2}$,

$u_{LR} = n \ln\left(1 + \frac{t_W^2}{n-1}\right)$, kde $t_W = \frac{\bar{x}-\mu_0}{S_{n-1}/\sqrt{n}}$
(pozn.: používáme směrodatnou odchylku S_{n-1}, tj. funkci `sd()`)

Vytvoříme funkce pro výpočet statistik na základě statistiky t_W, kde vstupem bude právě statistika t_W a rozsah náh. výběru n.

```
UW <- function(tw, n){
  uw <- tw ^ 2 * (n / (n - 1))
  return(uw)
}
```

Výsledky:
t_W = -2.8004105, u_W = 7.9162827, u_S = 7.3709506, u_{LR} = 7.6371306.

Funkce t.test()

V R lze pro výpočet t-statistiky testu o střední hodnotě použít rovněž funkci `t.test()`

```
t.test(skull.F, mu = 126.942)
```

```
	One Sample t-test

data:  skull.F
t = -2.8004, df = 106, p-value = 0.006067
alternative hypothesis: true mean is not equal to 126.942
95 percent confidence interval:
 124.7904 126.5741
sample estimates:
mean of x
 125.6822
```

Test pomocí kritického oboru

Waldův test: $t_{n-1}(1-\alpha/2), t_{n-1}(\alpha/2)$

Test poměrem věrohodnosti: $\chi^2_1(\alpha)$ (pozn.: v R zadáváme jako 1-alpha)

```
alpha <- 0.05
(qw <- qt(alpha / 2, n - 1))
```

```
[1] -1.982597
```

```
(qlr <- qchisq(1 - alpha, 1))
```

```
[1] 3.841459
```

Máme tedy kritické obory $W_W = (-\infty, -1.98) \cup (1.98, \infty)$ a $W_{LR} = (3.84, \infty)$.

Testy pomocí IS a p-hodnoty

Waldův IS máme ve výstupu funkce `t.test()` (vzorcem si vypočítejte samostatně jako DÚ).

```
dhW <- t.test(skull.F, mu = 126.942)$conf.int[1]
hhW <- ...
```

Pro věrohodnostní IS musíme sestrojit posloupnost pro μ přibližně mezi hodnotami $\bar{x}-s, \bar{x}+s$ o minimální délce 2000 (čím více bodů, tím přesnější máme hranice IS), vytvořit vektory hodnot test. statistik v této posloupnosti a porovnat je s kritickou hodnotou:

```
mu_i <- seq(m - s, m + s, length = ...)
tW_i <- (m - mu_i) / s * sqrt(n)
uW_i <- ... # statistika uW pro tW_i

dh_uW <- round(min(mu_i[uW_i < qlr]), 4)
hh_uW <- round(max(mu_i[uW_i < qlr]), 4)
# analogicky pro uS a uLR

## p-hodnoty
pW <- 2 * min(pt(tW, n - 1), 1 - pt(tW, n - 1))
p_uW <- 1 - pchisq(uW,  1)
p_uS <- ...
p_uLR <- ...
```

Nakonec sestavíme tabulku výsledků, kde budou pro jednotlivé statistiky jejich hodnoty, hranice IS, kritických oborů a p-hodnota testu.

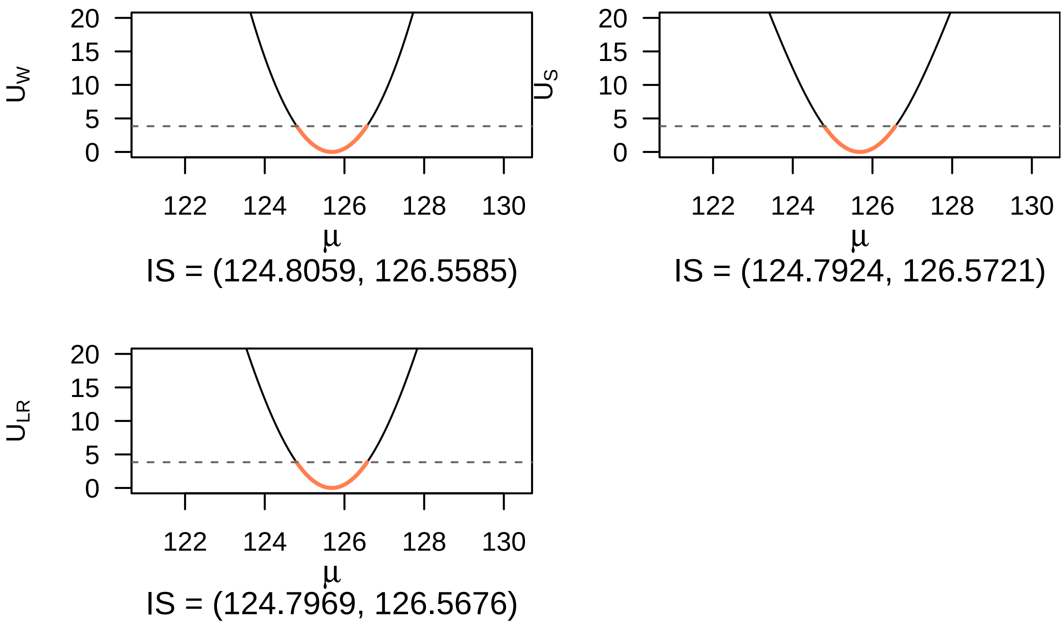
Tabulka výsledků

	statistika	W_{hh}	W_{dh}	IS_{dh}	IS_{hh}	p-hodnota
t_W	-2.800	-1.983	1.983	124.790	126.574	0.006
u_W	7.916	NA	3.841	124.806	126.558	0.005
u_S	7.371	NA	3.841	124.792	126.572	0.007
u_{LR}	7.637	NA	3.841	124.797	126.568	0.006

Interpretace: Mezi basion-bregmatickou výškou lebky žen starověké a novověké egyptské populace existuje statisticky významný rozdíl ($\mu - \mu_0 =$ -1.259757).

Grafy IS

```r
par(mar = c(5, 4, 1, 0), mfrow = c(2,2))
plot(mu_i, uW_i, type = 'l', ylim = c(0, 20), xlab = '',
      ylab = bquote(U[W]), las = 1)
lines(mu_i[uW_i < qlr], uW_i[uW_i < qlr], col = 'coral', lwd = 2)
abline(...) # seda carkovana cara v hodnote kvantilu
mtext(bquote(mu), side = 1, line = 2.2)
mtext(bquote(paste('IS = (', .(dh_uW), ', ', .(hh_uW), ')' )),
      side = 1, line = 3.3)
```



Příklad 2

Rozdělení testovacích statistik t_W a t_W^2

Nechť náhodný výběr X pochází z normálního rozdělení, t.j. $X \sim N(\mu, \sigma^2)$, kde $\mu = 150$ a $\sigma^2 = 30^2$. Rozsah náhodného výběru zvolte (a) $n = 10$; (b) $n = 100$. Pomocí simulační studie ověřte, že pro test o střední hodnotě μ, když σ^2 neznáme, platí za platnosti H_0 následující:

1. $T_W = \frac{(\bar{x}-\mu_0)}{s/\sqrt{n}} \sim t_{n-1}$;
2. $T_W^2 = \frac{(\bar{x}-\mu_0)^2}{s^2/n} \sim F_{1,n-1}$;

3. $U_W = t_W^2 \frac{n}{n-1} \sim \chi_1^2$;
4. $U_S = \dfrac{\frac{n t_W^2}{n-1}}{1 + \frac{t_W^2}{n-1}} \sim \chi_1^2$;
5. $U_{LR} = n \ln\left(1 + \frac{t_W^2}{n-1}\right) \sim \chi_1^2$.

Postup

Vygenerujte $M = 1000$ pseudonáhodných výběrů o rozsahu $n = 5$. Pro každý náhodný výběr vypočítejte hodnoty realizací testovacích statistik. Vytvořte histogramy těchto testovacích statistik a superponujte je křivkami příslušného rozdělení. Dále vytvořte animaci, pomocí které vizualizujete přesnost rozdělení testovacích statistik při různých rozsazích náhodného výběru. Animaci vytvořte pro měnící se $n \in \{5, 10, 15, 20, 25, 30, 35, 40, 50, 100, 250, 500\}$.

V R vytvoříme funkci TStat, která nám pro vstupní parametry rozdělení spočítá test. statistiky a v závislosti na typu statistiky (parametr `type`) nám vykreslí histogramy proložené křivkou přílušného rozdělení.

Funkce TStat

```
TStat <- function(mu, sigma, M = 1000, n, type = 'tW') {
  x <- rnorm(M * n, mu, sigma)
  X <- matrix(x, nrow = M)
  m <- apply(...) # vektor prumeru
  s <- apply(...) # vektor vyb. sd
  tW <- ... # tW

  if (type == 'tW') {
    X <- tW
    main <- expression(t[W])
    xlim = c(-7, 7)
    xx <- seq(-7, 7, length = 512)
    yy <- ... # hustota Student. rozdeleni
              # o n-1 stupnich volnosti
    }
  if (type == 'tW2') {...}
  # rozsah x pro tw^2 volte (0,15)
  if (type == 'uW') {
    X <- ...
    main <- ...
```

```
    xlim = c(0, 10)
    xx <- seq(0, 15, length = 512)
    yy <- ... # hustota prislusneho rozdeleni pro uW
      }
  ... ## doplnte pro uS a uLR
```

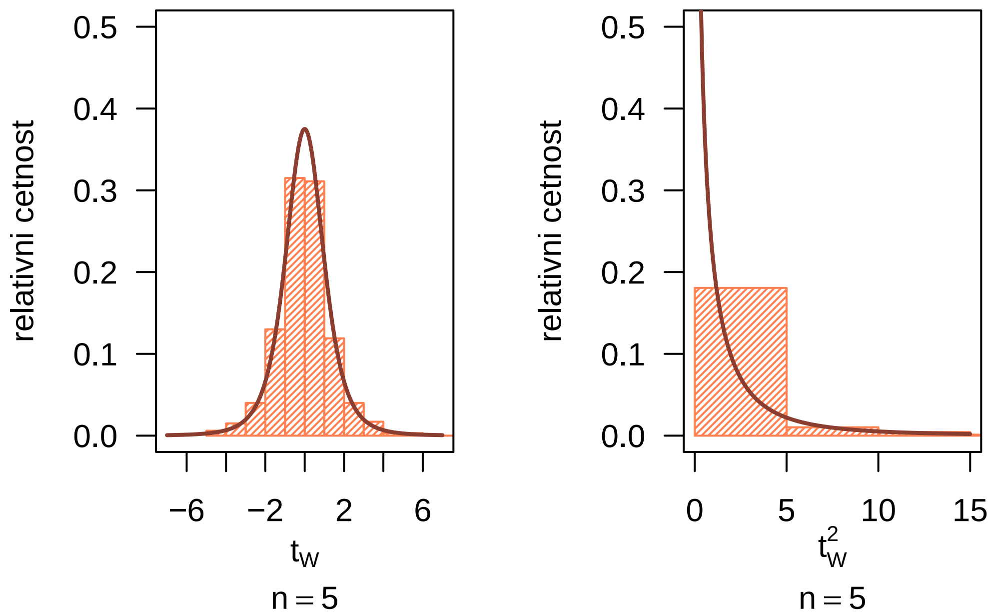
Tipy pro nastavení histogramu:

```
## pokracovani funkce TStat
## nastaveni histogramu
hist(...,  main = '', ylim = c(0, 0.5),
     xlim = xlim, breaks = 15)
  box(bty = 'o')
  ... # vykresleni krivky hustoty
  mtext(main, side = 1, line = 2.2)
  mtext(bquote(paste(n == .(n))), side = 1, line = 3.3)

}
```

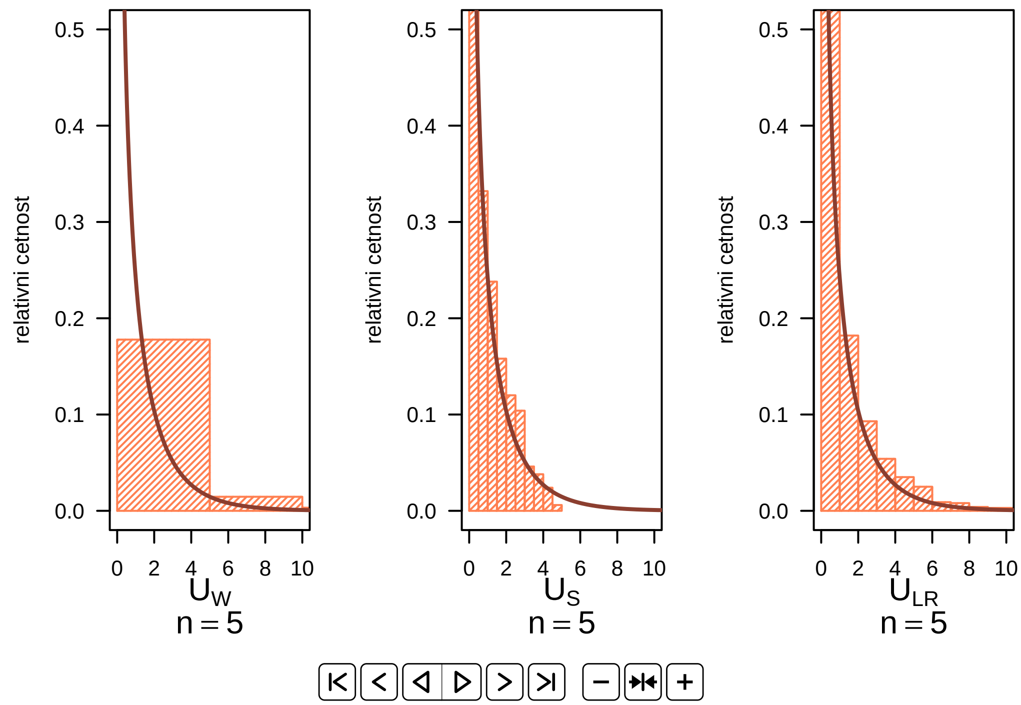
Výsledné animace

````
```{r}
#| fig-show: animate
#| eval: false
library(gifski)
n <- ...
par(mfrow = c(1, 2), mar = c(5, 5, 1, 1))
for (i in 1:length(n)) {
  TStat(mu = ..., sigma = ..., n = n[i],  type = 'tW')
  TStat(mu = ..., sigma = ..., n = n[i],  type = 'tW2')
}
```
````

Výsledná animace pro T_W a T_W^2


0.5
0.4
0.3
0.2
0.1
0.0
relativni cetnost
−6
−2
2
6
t_W
n = 5
0
5
10
15
t_W^2
n = 5

Animace pro věrohodnostní statistiky



Příklad 3

Porovnání testovacích statistik U_W, U_S a U_{LR}

Pomocí simulační studie porovnejte tvary těchto tří testovacích statistik a ukažte, že platí vztah $U_S < U_{LR} < U_W$.

Vygenerujte $M = 1000$ náhodných výběrů $X \sim N(0, 3)$ a pro každý výběr vypočítejte hodnoty testovacích statistik U_W, U_S a U_{LR} pro test nulové hypotézy
$H_0 : \mu = \mu_0 = 0$ oproti alternativní hypotéze $H_1 : \mu \neq \mu_0$.

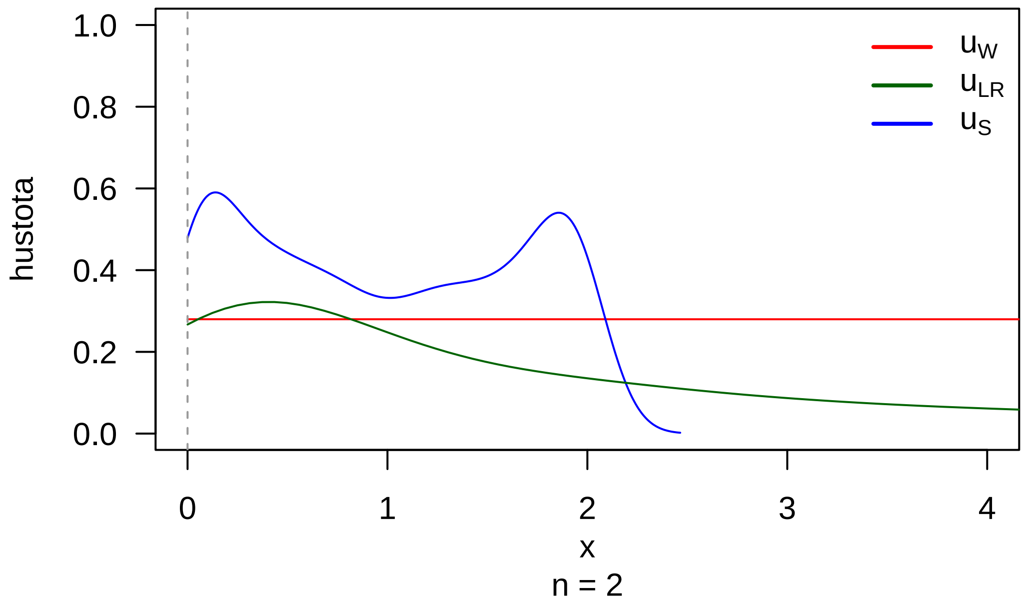
(A) Pro každou testovací statistiku následně najděte její jádrový odhad a křivky jádrového odhadu pro pevně zvolené n vykreslete do jednoho grafu. Vytvořte animaci zobrazující tvary křivek jádrových odhadů testovacích statistik při rostoucím rozsahu náhodných výběrů n. Rozsah výběrů n volte postupně $n = 2, 3, \dots, 9, 10, 15, 20, 25, 30, 35, 40, 50, 100, 250, 500$.

(B) Dále vypočítejte průměrné hodnoty testovacích statistik U_W, U_S a U_{LR} pro n = 5, 10, 100 a 1000 a zaneste je do jednoho grafu.

Funkce PorovnaniU

```
PorovnaniU <- function(n, M = 1000, mu0, mu, sigma, plot = T) {
  X <- t(replicate(M, rnorm(n, mu, sigma)))
  ... # vypocet m, s, tw, uW, uS, uLR jako u pr. 2

  prumery <- c(mean(uW), mean(uS), mean(uLR)) # predchystani pro B)
  if (plot == T) { # graf se vykresli pouze pro plot = T
    plot(uW, # priprava prazdneho grafu
      type = "n", ylim = c(0, 1), xlim = c(0, 4),
      xlab = "", ylab = "hustota", main = "", las = 1
    )
    lines(density(uW, from = 0), col = "red") # krivka jadroveho odhadu hustoty,
    # z definice nemuze byt statistika zaporna
    ... # dalsi 2 krivky, legenda
    abline(v = 0, lty = 2, col = "gray60") # pridani ref. krivky do 0
  }
  return(prumery = prumery)
}
```

Výsledná animace

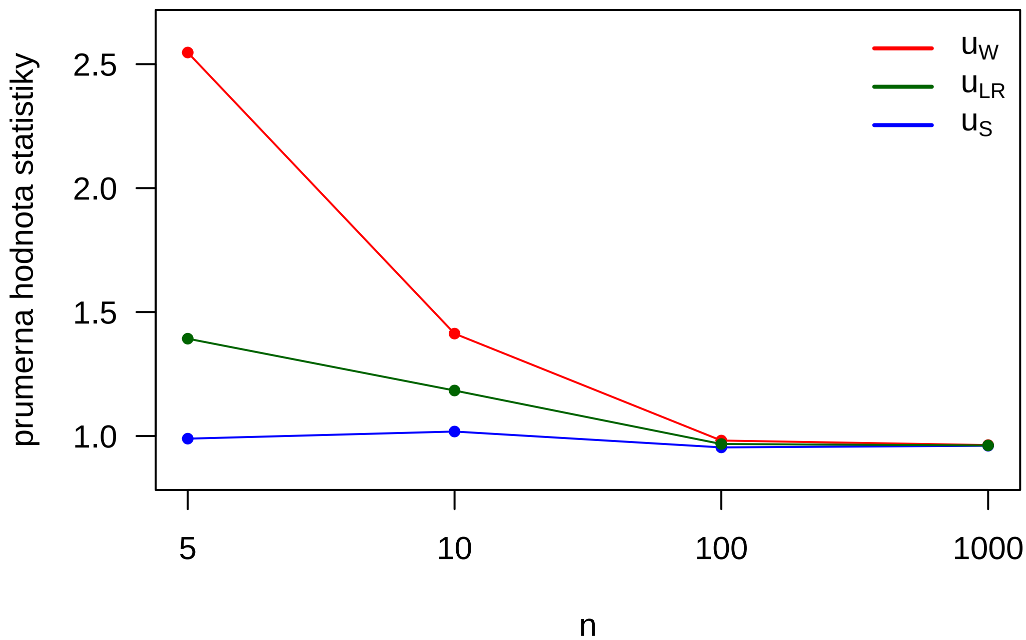


Z animace je patrné, že se statistiky asymptoticky blíží. Také si můžeme všimnout, že pro malé rozsahy náh. výběru statistika U_W selhává a není tedy pro testy o střední hodnotě při neznámém rozptylu v případech malých n vhodná.

B) graf průměrných hodnot statistik

```
# pro vypocet prumeru pouzijeme funkci PorovnaniU s nastavenim plot = F
m5 <- PorovnaniU(mu0 = 0, mu = 0, sigma = sqrt(3), n = 5, plot = F)
m10 <- ... # m100, m1000
m <- cbind(m5, m10, m100, m1000)
m_uW <- m[1, ] # prumery uW jsou v prvnim sloupci m
m_uS <- ... # prumery uS jsou ve druhem sloupci m
m_uLR <- ... # prumery uLR jsou ve tretim sloupci m

plot(1:4, m_uW, type = 'o', ylim = c(min(m_uS) - 0.1, max(m_uW) + 0.1),
     axes = F, ...)
# ylim nastavujeme podle minima nejmensi a maxima nejvetsi statistiky
# type = 'o' vykresli body spojene carami
```

```r
axis(1, at = c(1, 2, 3, 4), labels = c(5, 10, 100, 1000)) # zmena popisku osy x
# pridame dalsi krivky a legendu
```



Příklad 4

Hustota a distribuční funkce centrálního a necentrálního t-rozdělení

Nakreslete
(a) hustotu;
(b) distribuční funkci
jednoho centrálního a čtyř necentrálních t-rozdělení $t_{n-1,\lambda}$ ($\delta = \mu - \mu_0$ a $\lambda = \delta/(\sigma/\sqrt{n})$) do jednoho obrázku tak, aby byly odlišitelné barvou nebo typem čáry. Zvolte $\mu_0 = 0$, $\sigma = 1.4$, $n = 26$, $\delta = 0$, 0.5, 0.8, 1 a 1.2.

Postup v R

```r
delta <- c(...) # vektor delta
barvy <- c(...) # zvolime vektor 5 barev

plot(0, 0, type = 'n', xlab = 'x', ylab = 'hustota', main = '',
```

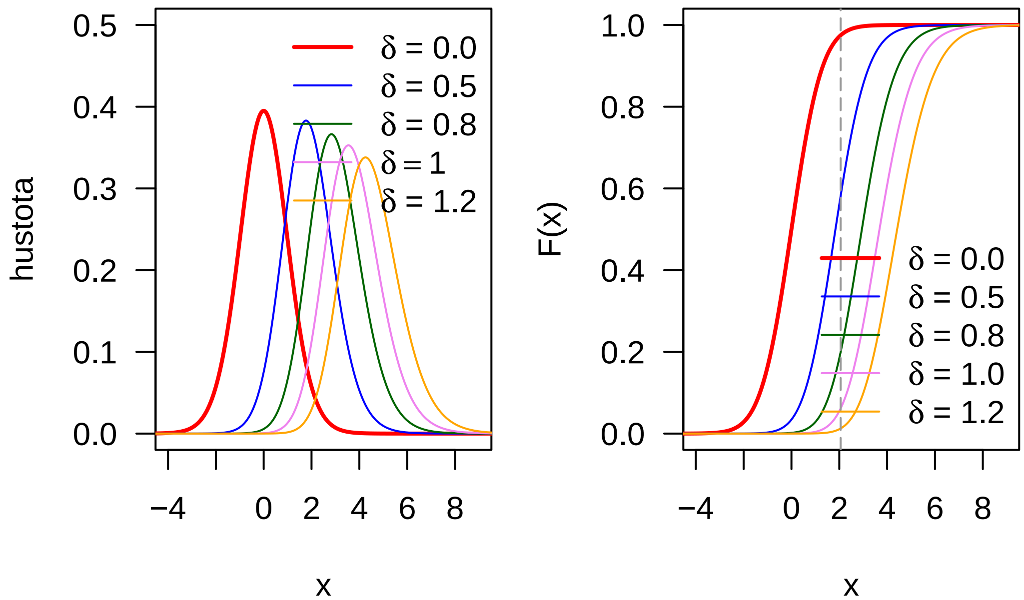
```r
     las = 1, xlim = c(-4, 9), ylim = c(0, 0.5)) # predchystani grafu

for (i in 1:length(delta)) {
  lambda <- ... # parametr necentrality pro delta[i]
  xfit   <- seq(-5, 10, length = ...)
  yfit   <- ...(..., ..., ncp = lambda)
  # hustota student. rozdeleni s n-1 stupni volnosti
  lines(..., ..., col = barvy[i])
}

## obdobne pro distribucni funkci
# nezapomente spravne nastavit ylim u distribucni fce
# v grafu distrib. fce cheme mit referencni caru
# v hodnote kvantilu t_n-1(1- alpha/2)
# ta by mela protinat krivku ve vysce 0.975
abline(v = qt(0.975, n), col = 'grey60', lty = 5)
```

Vytvoření legendy

Protože funkce `legend()` se přepisuje při každém zavolání, je nutné zadat ji až na konci mimo `for` cyklus (pokud bychom ji chtěli ponechat uvnitř cyklu, museli bychom nastavit postupné posouvání souřadnic, kde se má legenda vykreslit). K vykreslení symbolu δ s příslušnou hodnotou můžeme použít 2 způsoby:

```r
# 1. zpusob
legend('pozice', col = barvy,
       legend = c(expression(paste(delta,' = 0.0')), # rucni nastaveni hodnoty
                  ... # totez pro dalsi hodnoty
                  ))
# 2. zpusob
legend('pozice', col = barvy,
       legend = c(bquote(paste(delta == .(delta[1]))),
                  ... # analogicky pro dalsi hodnoty
                  ))
```

Graf hustoty a distribuční fce



Z grafu distribuční funkce lze pozorovat, jak necentralita ovlivňuje vztah kvantilu a distribuční fce. Dále lze z grafů pozorovat, že s měnící se hodnotou λ se mění nejen střední hodnota, ale i rozptyl Studentova necentrálního rozdělení.