MI-SPI 2017 – Domácí úkol č.2

*Vedoucí týmu: Lorenc Petr (lorenpe2, **107**)*
*Členové týmu: Liutova Oleksandra (liutoole, **107**)*
Datum: 11.5.2017

1. (1.5 bodu) Jednovýběrový t-test pro střední hodnotu:

1. **(1.5 bodu) Jednovýběrový t-test pro střední hodnotu:**

I. Oboustranný t-test pro střední hodnotu jednoho náhodného výběru provedeme v R příkazem `t.test`:

```
n = 20;
alpha = 0.01;
x = rnorm(n, mean=10.5, sd=1.3);

hypothesisTest = t.test(x, mu=10, conf.level = 1-alpha);
print(hypothesisTest); # Printing of the result is useful if you execute a script file
```

Proveďte následující výpočty a **porovnejte své výsledky s výstupem** předchozího příkazu `t.test` (bez porovnání a diskuze ztratíte body):

a. Spočtěte příslušný oboustranný 99% konfidenční interval pro střední hodnotu 'mu'. Kvantily a kritické hodnoty Studentova t-rozdělení získáte pomocí

```
quantile = qt(probability, degreesOfFreedom)
criticalValue = qt(probability, degreesOfFreedom, lower.tail = FALSE)
```

b. Pomocí tohoto intervalu otestujte H_0: mu = 10 proti oboustranné alernativě H_A: mu <> 10. Vysvětlete, jaká je pravděpodobnost, že vaše rozhodnutí je chybné.

c. Spočtěte hodnotu testové T-statistiky. Porovnejte její hodnotu s příslušnou kritickou hodnotou a potvrďte tak své rozhodnutí z předchozího bodu.

d. **Extra 1/2 bodu:** Při jakém nejnižším možném 'alpha' byste H_0 mohli zamítnou? Ukažte přesně, jak využijete příkazu `pt(…)` s parametrem `lower.tail = FALSE` či `TRUE`. Je získaná hodnota nižší než alpha? Jak tato hodnota souvisí s předchozím výstupem příkazu `t.test`? Jak souvisí s rozhodnutím testu?

II. Jednostranný t-test získáme pomocí parametru `alternative` příkazu `t.test`:

```
t.test(x, mu=10, alternative = "greater", conf.level = 1-alpha);
t.test(x, mu=10, alternative = "less", conf.level = 1-alpha);
```

Vyberte, který z jednostranných testů je vhodnější pro naši situaci a zopakujte všechny kroky z předchozího bodu. Použijte H_0: mu = 10 a popište přesně, jak a proč jste zvolili alternativu H_A. Zdůvodněte přesně, který jednostranný konfidenční interval jste zvolili, a proč.

1.I.

--
hypothesisTest = t.test(x, mu=10, conf.level = 1-alpha);
--
vraci
--
data: x
t = 1.6851, df = 19, p-value = 0.1083
alternative hypothesis: true mean is not equal to 10
99 percent confidence interval:
9.645957 11.368840
sample estimates:
mean of x
10.5074
--
Takze nasi hypotezu H0 ze se prumery rovnaji 10 nemuzeme zamitnout (s moznou chybou 99%)

Ke stejnemu vysledku dojdeme i pokud budemem pocitat "rucne":
--

```
# oboustranna alternativa na 99% -> 1% rozdelit na obe strany
quantile = qt(0.995, n-1)

#Stred interavlu tedy prumer
xmean = mean(x);
#odchylka
stdDev = sd(x);
sqrtn = sqrt(n);

# z prednasky 17/46
intv = quantile*stdDev/sqrtn

confint = c(xmean - intv, xmean + intv)
print(confint)
```

```
t.test(x, mu=10, conf.level = 1-alpha);
print("Nas interval je")
print(confint)
print("Nase stredni hodnota je")
print(xmean)
```
--
vrati
--
```
[1] "Nas interval je"
[1]  9.645957 11.368840
[1] "Nas prumer je"
[1] 10.5074
```
--
coz je presne co jsme cekali a co nam take vratila funkce t.test ... testuje jestli je nase H0 v intervalu od 9.645957 do 11.368840 a to je.
Ted se pokusime spocitat T-statistiku k tomu pouzijeme vzorce z prednasky 17 slide 46
--
```
testT = (mean(x) - 10)/sqrt(var(x)/n)
intertvalT=c(-qt(.995, n-1), +qt(.995, n-1)) # oboustrany takze 0.5 procenta na kazde strane aby to bylo na tech 99 procent
```
--
vrati
--
```
> print(testT)
[1] 1.685123
> print(intertvalT)
[1] -2.860935  2.860935
```
--
a protoze **T-statistika je v rozmezi hodnot tak jsme potvrdili to co uz jsme rekli 2x a tj. ze H0 nezamitame (vse samozrejmen s pravdepodobnosti s 99%)**

1.II.

zamitnuti je silnejsi (da nam aspone jakou informaci) tak volime vetsi protoze apriori vime ze hodnota je 10.5 a my testujeme proti 10 tak hypoteza je ze to je 10 a testuje oproti alternative ze to je vice nez 10 (tj pokud zamitneme nulovou hypotezu tak ve prospech alternativi)
--
```
greater=t.test(x, mu=10, alternative = "greater", conf.level = 1-alpha);
```
--
vrati stejne cisla jako
--
```
criticalValue <- qt(.99,n-1);
xmean <- mean(x);
stdDev <- sd(x);
sqrtn <- sqrt(n);
intv <- criticalValue*stdDev/sqrtn

# z prednasky 17/66
confint = c(xmean - intv, Inf  )
print(greater)
print(confint) # pro kontrolu
in_interval(mu, confint)
#true -> H0 nezamitame

testT = (mean(x) - 10)/sqrt(var(x)/(n-1))
nint=c(-qt(.990, n-1), Inf)
print(nint)
in_interval(testT, nint)
#true ->H0 nezamitame .
```
--
tj vrati
--
t = 1.6851, df = 19, p-value = 0.05416
alternative hypothesis: true mean is greater than 10
99 percent confidence interval:

9.742748   Inf
sample estimates:
mean of x
  10.5074
--
odkud je videt ze nulovou hypotezu ze je stredni hodnota rovna 10 nemuzeme zamitnout. Bohuzel se tim teda nic nedokazuje (vse na hladine mozne chyby 99%)

2. **(1.5 bodu) Párový a dvouvýběrové t-testy pro porovnání středních hodnot:**

I. **Párový** t-test používáme pro porovnání středních hodnot veličin *X* a *Y*, pro které jsme napozorovali výběr nezávislých **párů** (x_i, y_i). Test nulové hypotézy H_0: $mu_X = mu_Y$ proti jednostranné alernativě H_A: $mu_X < mu_Y$ můžeme provést následovně:

```
n = 20;
alpha = 0.01
x = rnorm(n, mean=10, sd=1)
error = rnorm(n, mean=0.5, sd=0.8306624)
y = x + error

t.test(x, y=y, paired = TRUE, alternative = "less", conf.level = 1-alpha)
```

Všimněte si, že x_i a $y_i = x_i + error_i$ nejsou navzájem nezávislé, ale (x_i, y_i) jsou páry z nezávislých opakování experimentu.

a. Vyhodnoťte výstup z předchozího příkazu a otestujte H_0: $mu_X = mu_Y$ proti H_A: $mu_X < mu_Y$. Vysvětlete, jaká je pravděpodobnost, že vaše rozhodnutí je chybné.

b. Spočtěte rozdíly `diff = x - y` a otestujte nulovou hypotézu H_0: $mu_{Diff} = 0$ proti příslušné alternativě. Popište přesně jak a proč jste zvolili alternativu H_A. Porovnejte tento test s testem z předchozího bodu a diskutujte své závěry.

II. **Dvouvýběrový** t-test používáme pro porovnání středních hodnot veličin *X* a *Y* na základě **dvou *nezávislých* výběrů** dat. Výběry nemusí být stejně velké. Pokud *X* a *Y* mají **stejné rozptyly** (variance), pak použijeme příkaz `t.test` s parametry `paired = FALSE` a `var.equal = TRUE`:

```
n1 = 20;
n2 = 25;
alpha = 0.01
x=rnorm(n1, mean=10, sd=1.3)
y=rnorm(n2, mean=11.25, sd=1.3)

t.test(x, y=y, paired = FALSE, var.equal = TRUE, conf.level = 1-alpha)
```

a. Modifikujte předchozí příkaz pro test nulové hypotézy H_0: $mu_X = mu_Y$ proti jednostranné alernativě H_A: $mu_X < mu_Y$. Vyhodnoťte výstup z modifikovaného příkazu a otestujte H_0 proti H_A. Vysvětlete, jaká je pravděpodobnost, že vaše rozhodnutí je chybné.

b. Pomocí vzorců z přednášky spočtěte testovací statistiku 't' a stupně volnosti 'df' (degrees of freedom). Porovnejte své výsledky s výstupem předchozího příkazu `t.test`. Spočtěte bud příslušnou p-value či kritickou hodnotu t-rozdělení a potvrďte výsledek testu z předchozího bodu.

III. Pokud *X* a *Y* mají **rozdílné rozptyly** (variance), pak pro dvouvýběrový t-test v příkazu `t.test` změníme parameter `var.equal` na `FALSE`:

```
n1 = 20;
n2 = 25;
alpha = 0.01
x=rnorm(n1, mean=10, sd=1.3)
y=rnorm(n2, mean=11.28, sd=1.2)

t.test(x, y=y, paired = FALSE, var.equal = FALSE, conf.level = 1-alpha)
```

--
print(t.test(x, y=y, paired = TRUE, alternative = "less", conf.level = 1-alpha))
#zamitame hypotezu HO protoze vysel interval (-Inf -0.1216135) *a tam nula nepatri, pravdepodobnost chyby je 1 procento*

#2.I.b
diff = x - y
# prevedeno na priklad vyse s jednovyberovym t-testem
t.test(diff, mu=0, alternative = "less", conf.level = 1-alpha);
# H0 ze diff ma prumer v 0 zamitame - shoduje se s vysledkem vyse
--
#2 II
n1 = 20;
n2 = 25;
alpha = 0.01
x=rnorm(n1, mean=10, sd=1.3)
y=rnorm(n2, mean=11.25, sd=1.3)

t.test(x, y=y, paired = FALSE, var.equal = TRUE, conf.level = 1-alpha)
--
H0 ze maji stejne stredni **hodnoty nezamitame s pravdepodobnosti 99%, protoze t.test vyse vratil**

--
t = -2.2211, df = 43, p-value = 0.03166
alternative hypothesis: true difference in means is not equal to 0
99 percent confidence interval:
 -1.7944671 0.1729991
sample estimates:

mean of x mean of y
10.20239 11.01312
--
a 0 patri do konfidencniho intervalu.

--
print(t.test(x, y=y, paired = FALSE, var.equal = TRUE, conf.level = 1-alpha, alternative="less"))
 # H0 ze maji stejne rozplyly nezamitame ve prospech alternativy ze prumer rozdilu je mensi nez 0 s moznosti chyby na 99%
>>
t = -2.2211, df = 43, p-value = 0.01583
alternative hypothesis: true difference in means is less than 0
99 percent confidence interval:
-Inf 0.07121602
sample estimates:
mean of x mean of y
10.20239 11.01312
--
Degree of fredom najdeme v 18 prednasce slide 28 a **"df=m+n-2"** a pote odhadnuty rozptyl je **"sxy2 = ((sx*(n-1)) + (sy*(m-1))) / df"** tudiz smerodatna odchylka je **"sxy = sqrt(Sxy2)"** ze slidu 29 ziskame vzorec pro T hodnotu jako **"T= ((mean(x)-mean(y)) / (sxy*sqrt((1/n)+(1/m))))".** Pro zjisteni p-hodnoty ma R funkci pt a ta nam vratila stejne hodnoty jako jsou vyse. Tj p_**value** je vetsi nez alpha, tudiz **hypotezu h0 nezmitame** (kdyby lezela nalevo tak zamitame ale protoze testuje alternativu ze je mensi tak to ze je napravo je pro nas indikator ze nemuzeme zamitnout)

2 III
Maji rozdilne roztyly tak musime pouzit vzorecky z prednasky 18 slide 29 dole pro sigma1 != sigma2 kde dostaneme "**upper_part = ((sx/n1 + sy/n2)^2)** a **down_part = (((sx/n1)^2 / (n1-1)) + ((sy/n2)^2 / (n2-1)))**" potom degree of freadon je "**df = upper_part / down_part** ", smerodatna odchylak "**sxy = sqrt(sx/n1 + sy/n2)** "a T-hodnota"**T = (mean(x) - mean(y))/sxy**" podle vzorcu z tohoto slidu 29

Ted to tedy vyjde
--
t.test(x, y=y, paired = FALSE, var.equal = FALSE, conf.level = 1-alpha)
 # H0 ze maji stejne rozplyly zamitame ve prospech alternativy

#opakovat jako v predchozim bode ale s tim rozdilem ze rozplyt1 se nerovna rozptyl2
t.test(x, y=y, paired = FALSE, var.equal = FALSE, conf.level = 1-alpha, alternative="less")
 # H0 ze maji stejne rozplyly zamitame ve prospech alternativy ze prumer rozdilu je mensi nez 0 s moznosti chyby na 99%
--
a po provedeni vzorcu vyse dojdeme ke stejnemu vysledku
--
t = -3.5515, df = 40.802, p-value = 0.0004914
alternative hypothesis: true difference in means is less than 0
99 percent confidence interval:
-Inf -0.4110786
sample estimates:
mean of x mean of y
10.03112 11.32288
--
tj ze muzeme zamitnou s pravdepodobnosti chyby 99%

3. **(2 body) Praktické využití t-testů:**

I. Pro ilustraci praktického využití t-testů použijeme algoritmu quick sort implementovaného výpočetním systémem R.

a. V R máme k dispozici dvě verze quick sortu — příkaz `sort` s parametrem `method` buď „`shell`" (varianta Sedgewickovy verze), nebo „`quick`" (Singletonův quicksort).

b. Autoři tvrdí, že pro velké množiny numerických dat je Singletonův quick sort o něco rychlejší. Toto bude naše pracovní hypotéza.

c. Ověřte si rychlost svého počítače pomocí kódu

```
sequenceLength = 2000000;
x = runif(sequenceLength, 0, 100)
print(system.time(sort(x)))
```

Nastavte si parameter `sequenceLength` tak, aby čas v kategorii 'user' byl v rozsahu 0.25 - 0.75 sekundy.

II. Vygenerujte L*40 náhodných stejně dlouhých číselných sekvencí a změřte doby jejich seřazení. Každá sekvence bude seřazena oběma algoritmy. Např.

```
sampleSize = L*40;
time1 = time2 = numeric(sampleSize); # Declare an array
for(i in 1:sampleSize){
        x = runif(sequenceLength, 0, 100); # Generate the sequence to be sorted
        # Measure sort times. The user-space time is at system.time(...)[1]
        # Inside system.time we must use x1 <- value and not x = value. The latter syntax is reserved for parameters.
        time1[i] = system.time(x1 <- sort(x, method = "quick"),  gcFirst = TRUE)[1];
        time2[i] = system.time(x2 <- sort(x, method = "shell"), gcFirst = TRUE)[1];
}
```

a. Na hladině `alpha = K/100` otestujte, zda naměřená data poskytují statistickou evidenci pro naší pracovní hypotézu z předchozího bodu.

b. Popište přesně jak a proč jste zvolili nulovou hypotézu H_0 a alternativu H_A.

c. Zdůvodněte přesně, který t-test jste použili a proč.

III. Zopakujte předchozí bod pro oddělená měření, kdy každý algoritmus testován na své vlastní a odlišné sadě číselných sekvencí: L*40 a L*35 sekvencí. Např.

```
for(i in 1:sampleSize){
        x = runif(sequenceLength, 0, 100); # Generate the sequence to be sorted
        time1[i] = system.time(x1 <- sort(x, method = "quick"),  gcFirst = TRUE)[1];
}
sampleSize2 = L*35;
for(i in 1:sampleSize2){
        x = runif(sequenceLength, 0, 100); # Generate the sequence to be sorted
        time2[i] = system.time(x2 <- sort(x, method = "shell"), gcFirst = TRUE)[1];
}
```

IV. Porovnejte výsledky obou experimentů. Pokud se odlišují, vysvětlete proč.

Po provedeni napocitani casu budume testovat ze podle **zadani je Singletonův quick sort je o něco rychlejší :**

--

#jako Ho bereme ze bezi stejnou dobu
#jako Ha volime ze Singletonův quick sort je o něco rychlejší tj time1 - time2 bude mensi nez 0
diff=time1-time2
alfa=K/100

print(t.test(diff, mu=0, alternative = "less", conf.level = 1- alfa))
 # H0 zamitame ve prospech alternativy ze Singletonův quicksort je rychlejsi s pravdepodobnosti chyby 4%
 # jako alternativu jsme zvolili to cemu vice verime (nasi pracovni hypotezu ze Singletonův quick sort je o něco rychlejší)

--

vrati

--

t = -50.514, df = 239, p-value < 2.2e-16
alternative hypothesis: true mean is less than 0
96 percent confidence interval:
-Inf -0.2214318
sample estimates:
mean of x
-0.2294167

--

coz potvrzuje ze je opravdu rychlejsi s pravdepodobnosti chyby jsou 4 procenta

3.II

--

print(t.test(time1, y=time2, paired = FALSE, var.equal = FALSE, conf.level = 1-alpha, , alternative = "less"))
 # H0 zamitame ve prospech alternativy ze Singletonův quicksort je rychlejsi s pravdepodobnosti chyby 4%
 # jako alternativu jsme zvolili to cemu vice verime (nasi pracovni hypotezu)

--