MI-SPI 2015 − Domácí úkol č.2

Vedoucí týmu: Ondřej Paška (paskaond, 109)

Členové týmu: Tomáš Sušánka (susantom, 109), Jan Tvrdík (tvrdija4, 109)

Datum: 2.5.2015

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

Příkazy R dle instrukcí:

n = 20;

alpha = 0.01;

x = rnorm(n, mean=10.5, sd=1.3);

print(x);

hypothesisTest = t.test(x, mu=10, conf.level = 1-alpha);

print(hypothesisTest);

Výstup:

10.855855 8.990733 13.318086 10.817092 10.462166 8.758253 12.649601 11.177882 10.809809 9.653954 8.055347 8.549485 9.753337 9.668844 12.167107 9.253401 10.522409 11.633676 9.697370 11.234175

One Sample t-test

data: x

**t = 1.2863**, df = 19, p-value = 0.2138

alternative hypothesis: true mean is not equal to 10

99 percent confidence interval:

**9.508574 11.294284**

sample estimates:

mean of x

10.40143

Vygenerovali jsme vzorek 20 hodnot s normálním rozdělením. Pomocí t-testu jsme stanovili konfidenční interval pro střední hodnotu, konkrétně ve kterém se na **99% hladině spolehlivosti** bude nacházet střední hodnota.

Hypotézu, že na základě tohoto konfidenčního intervalu nelze vyvrátit.

* + 1. Spočtěte příslušný oboustranný 99% konfidenční interval pro střední hodnotu 'mu'.

Příkazy R dle instrukcí:

probability = 1 - (alpha / 2)

degreesOfFreedom = n-1

quantile = qt(probability, degreesOfFreedom)

# slide 50 lecture 17

marginError = (sd(x)/sqrt(n)) \* quantile

intervalLowerBound = mean(x) - marginError

intervalUpperBound = mean(x) + marginError

print(sprintf("Confidence interval = (%.3f, %.3f)", intervalLowerBound, intervalUpperBound));

Výstup:

Confidence interval = **(9.509, 11.294)**

Konfidenční interval počítaný ručně nám vyšel **stejně jako výše**. Vyjdeme z bodového odhadu , odhadu pomocí . Pomocí najdeme příslušný kvantil studentovat rozdělení odpovídající zadané a stupňům volnosti a převedeme ho ze směrodatných odchylek studentova rozdělení na směrodatnou ochylku

* + 1. Pomocí tohoto intervalu otestujte H0: mu = 10 proti oboustranné alernativě HA: mu <> 10. Vysvětlete, jaká je pravděpodobnost, že vaše rozhodnutí je chybné.

Příkazy R dle instrukcí:

## 1 Ib

mu = 10

if ((mu > intervalLowerBound) && (mu < intervalUpperBound)) {

print("H0 is not rejected")

} else {

print("H0 is rejected")

}

Výstup:

**H0 is not rejected**

Hodnota 10 spadá do nalezeného konfidenčního intervalu, takže **hypotézu nelze zamítnout**. Chyba v tomto případě odpovídá nezamítnutí hypotézy přestože ve skutečnosti by neplatila. Jedna se o tzv. c*hybu druhého druhu*. Rozhodně nemůžeme tvrdit, že hypotéze platí.

* + 1. Spočtěte hodnotu testové T-statistiky. Porovnejte její hodnotu s příslušnou kritickou hodnotou a potvrďte tak své rozhodnutí z předchozího bodu.

Příkazy R dle instrukcí:

tStatisticValue = (mean(x) - mu) / (sd(x) / sqrt(n))

print(sprintf('T = %.3f', tStatisticValue));

if (abs(tStatisticValue) > quantile) {

print("H0 is rejected by T-statistic")

} else {

print("H0 is not rejected by T-statistic")

}

Výstup:

**T = 1.286**

**H0 is not rejected** by T-statistic

I hodnota t-statistiky nám vyšla **stejně jako výše** a podporuje **nazamítnutí**  Hodnota t-statistiky je vyjadřuje, jak se liší bodový odhad na základě náhodného vyběru od tvrzení dané hypotézou v jednotkách směrodatné odchylky, tj. kolik směrodatných odchylek mají mezi sebou.

* + 1. Při jakém nejnižším možném 'alpha' byste H0 mohli zamítnou?

Příkazy R dle instrukcí:

alpha2 = 2 \* (1 - pt(tStatisticValue, degreesOfFreedom));

Výstup:

**0.2137991**

Použijeme inverzní funkce ke , tj. s cílem splnit rovnici

Získaná hodnota je **výrazně větší, než původní alpha**, takže výsledný konfidenční interval je menší, takže hodnota 10 leží mimo něj. Tato hodnota **odpovídá hodnotě p-value** v původním výstupu t.test.

To není náhoda, P-value nám totiž říká, jaké je pravděpodobnost, že dostaneme náhodný výběr alespoň tak extrémní jako ten, který máme, za předpokladu, že platí. Pokud se srovná s touto pravděpodobností extrému, tak vlastně říkáme, že nám to nevadí a akceptujeme tak odpovídající riziko omylu.

* 1. Jednostranný t-test získáme pomocí parametru alternative příkazu t.test:

Příkazy R dle instrukcí:

print("First one-sided tests:")

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

print("Second one-sided tests:")

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

Výstup:

One Sample t-test

data: x

t = **1.2863**, df = 19, p-value = 0.1069

alternative hypothesis: true mean is greater than 10

99 percent confidence interval:

**9.608894 Inf**

sample estimates:

mean of x

10.40143

One Sample t-test

data: x

t = **1.2863**, df = 19, p-value = 0.8931

alternative hypothesis: true mean is less than 10

99 percent confidence interval:

**-Inf 11.19396**

sample estimates:

mean of x

10.40143

Ani jeden z intervalů nám **neumožňuje zamítnout hypotézu** , takže je vybereme interval, ke kterému příslušší nižší p-value – volíme

* + 1. Spočtěte příslušný jednostranný 99% konfidenční interval

Příkazy R dle instrukcí:

probability = 1 - alpha # 0.99

quantile = qt(probability, degreesOfFreedom)

marginError = (sd(x)/sqrt(n)) \* quantile

intervalLowerBound = mean(x) - marginError

intervalUpperBound = +Inf

print(sprintf("Confidence interval = (%.3f, %.3f)", intervalLowerBound, intervalUpperBound));

Výstup

Confidence interval = (9.609, Inf)

Interval **opět vyšel stejně** jako při použití příkazu t-test.

* + 1. Pomocí tohoto intervalu otestujte H0: mu = 10 proti jednostranné alternativě HA: mu < 10. Vysvětlete, jaká je pravděpodobnost, že vaše rozhodnutí je chybné.

Příkazy R:

if ((mu > intervalLowerBound) && (mu < intervalUpperBound)) {

print("H0 is not rejected")

} else {

print("H0 is rejected")

}

Výstup:

**H0 is not rejected**

Hypotéza opět **není zamítnuta**. Možná chyba tohoto rozhodnutí je **2. řádu.**

* + 1. Spočtěte hodnotu testové T-statistiky. Porovnejte její hodnotu s příslušnou kritickou hodnotou a potvrďte tak své rozhodnutí z předchozího bodu

Příkazy R:

tStatisticValue = (mean(x) - mu) / (sd(x) / sqrt(n))

if (abs(tStatisticValue) > quantile) {

print("H0 is rejected by T-statistic")

} else {

print("H0 is not rejected by T-statistic")

}

Výstup:

**1.286281**

**H0 is not rejected** by T-statistic

T-statistika opět **vyšla stejně** jako při použití t-test. Hypotéza opět **není zamítnuta**.

* + 1. Při jakém nejnižším možném 'alpha' byste H0 mohli zamítnou?

Příkazy R:

alpha3 = 1 - pt(tStatisticValue, degreesOfFreedom);

Výstup:

**0.1068996**

Hodnota **opět odpovídá p-value**.

1. Párový a dvouvýběrové t-testy pro porovnání středních hodnot:
   1. Párový t-test
      1. Vyhodnoťte výstup z předchozího příkazu a otestujte H0: muX = muY proti HA: muX < muY. Vysvětlete, jaká je pravděpodobnost, že vaše rozhodnutí je chybné.

Příkazy R dle instrukcí:

n = 20;

alpha = 0.01;

x = rnorm(n, mean=10, sd=1);

error = rnorm(n, mean=0.5, sd=0.8306624);

y = x + error;

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

## 2 Ia

print(tTestPair)

* + - * Výstup:

Paired t-test

data: x and y

t = -2.7241, df = 19, p-value = 0.006734

alternative hypothesis: true difference in means is less than 0

99 percent confidence interval:

-Inf -0.04023544

sample estimates:

mean of the differences

-0.5936044

* Provedli jsme test hypotézy H0: muX = muY proti hypotéze HA: muX < muY.
* Vidíme, že konfidenční interval neobsahuje 0, neboť je roven (-inf, -0.04023544). Z toho plyne, že můžeme hypotézu H0 zamítnout. To také potvrzuje velmi nízká hodnota p-value = 0.006734.
* Pravděpodobnost, že naše rozhodnutí je chybné, aneb že jsme zamítli H0, i když je H0 pravdivé, je rovna chybě prvního typu a pravděpodobnost této chyby je rovna p-value, tedy 0.6734%.
  + 1. Spočtěte rozdíly diff = x – y a otestujte nulovou hypotézu H0: muDiff = 0 proti příslušné alternativě. Popište přesně jak a proč jste zvolili alternativu HA. Porovnejte tento test s testem z předchozího bodu a diskutujte své závěry.

Příkazy R dle instrukcí:

## 2 Ib

diff = x – y

test = t.test(diff, y=y, paired=TRUE, alternative = "less", conf.level = 1-alpha, mu=0)

print(test)

* + - * Výstup:

Paired t-test

data: diff and y

t = -20.4532, df = 19, p-value = 1.059e-14

alternative hypothesis: true difference in means is less than 0

99 percent confidence interval:

-Inf -9.175537

sample estimates:

mean of the differences

-10.47628

* Protože jsme v minulém bodě testovali H0: muX = muY proti HA: muX < muY a v v tomto úkolu máme H0: muDiff = 0, volíme analogicky HA: muDiff < 0.
* Ze získaných dat vidíme extrémně nízkou p-value (1.059e-14). H0 tedy můžeme opět zamítnout.
* V porovnání s předešlým bodem zjistíme, že výsledek zamítnutí je stejný, máme však výrazně nižší p-value a tudíž také nižší pravděpodobnost chyby.
  1. Dvouvýběrový t-test

Příkazy R dle instrukcí:

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)

* + - * Výstup:

Two Sample t-test

data: x and y

t = -1.2522, df = 43, p-value = 0.2173

alternative hypothesis: true difference in means is not equal to 0

99 percent confidence interval:

-1.7818544 0.6513663

sample estimates:

mean of x mean of y

10.39239 10.95763

* + 1. Modifikujte předchozí příkaz pro test nulové hypotézy H0: muX = muY proti jednostranné alernativě HA: muX < muY. Vyhodnoťte výstup z modifikovaného příkazu a otestujte H0 proti HA. Vysvětlete, jaká je pravděpodobnost, že vaše rozhodnutí je chybné.

Příkazy R dle instrukcí:

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

* + - * Výstup:

Two Sample t-test

data: x and y

t = -1.2522, df = 43, p-value = 0.1086

alternative hypothesis: true difference in means is less than 0

99 percent confidence interval:

-Inf 0.5254883

sample estimates:

mean of x mean of y

10.39239 10.95763

* Vidíme, že konfidenční interval obsahuje 0, neboť je roven (-inf, 0.5254883). Z toho plyne, že nemůžeme hypotézu H0 zamítnout. To potvrzuje velmi vysoká hodnota p-value, která je rovna 10,86%.
* Pravděpodobnost, že naše rozhodnutí je chybné, aneb že jsme nezamítli H0, i když je H0 nepravdivé, je rovna chybě druhého typu a pravděpodobnost této chyby je neznámá.
  + 1. 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.

Příkazy R dle instrukcí:

## 2 IIb

s2x = sum( (x - mean(x))^2 ) / (length(x)-1)

s2y = sum( (y - mean(y))^2 ) / (length(y)-1)

Sxy = sqrt( ((length(x)-1)\*s2x + (length(y)-1)\*s2y) / (length(x)+length(y) -2) )

TStat = (mean(x) - mean(y)) / (Sxy \* sqrt(1/length(x) + 1/length(y)))

print(TStat)

df = length(x) + length(y) -2

print(df)

tCriticalValue = qt(1-alpha, df, lower.tail = TRUE)

pVal = pt(TStat, df = df)

print(pVal)

* + - * Výstup:

[1] -1.25216

[1] 43

[1] 0.1086399

* Spočtené hodnoty jsou: t = -1.25216, df = 43 a p-value = 0.1086399. V minulém bodě nám vyšlo t = -1.2522, df = 43 a p-value = 0.1086. Je zřejmé, že hodnoty jsou téměř stejné, jejich malé rozdíly jsou dané pouze chybami způsobenými během dělení.
  1. Pokud X a Y mají rozdílné rozptyly (variance), pak pro dvouvýběrový t-test v příkazu t.test změníme parameter var.equal na FALSE:

Příkazy R dle instrukcí:

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)

* + - * Výstup:

Welch Two Sample t-test

data: x and y

t = -2.7964, df = 37.766, p-value = 0.008082

alternative hypothesis: true difference in means is not equal to 0

99 percent confidence interval:

-2.38728952 -0.03640032

sample estimates:

mean of x mean of y

9.974145 11.185990

* + 1. Modifikujte předchozí příkaz pro test nulové hypotézy H0: muX = muY proti jednostranné alernativě HA: muX < muY. Vyhodnoťte výstup z modifikovaného příkazu a otestujte H0 proti HA. Vysvětlete, jaká je pravděpodobnost, že vaše rozhodnutí je chybné.

Příkazy R dle instrukcí:

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

* + - * Výstup:

Welch Two Sample t-test

data: x and y

t = -2.7964, df = 37.766, p-value = 0.004041

alternative hypothesis: true difference in means is less than 0

99 percent confidence interval:

-Inf -0.1591313

sample estimates:

mean of x mean of y

9.974145 11.185990

* Stejně jako v předešlých bodech se podívame na konfidenční interval a zjistíme,že neobsahuje 0, neboť je roven (-inf, -0.1591313). Z toho plyne, že můžeme hypotézu H0 zamítnout.
* Pravděpodobnost, že naše rozhodnutí je chybné je rovno 0.004041, tedy 0.4041%.
  + 1. 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.

Příkazy R dle instrukcí:

s2x = sum( (x - mean(x))^2 ) / (length(x)-1)

s2y = sum( (y - mean(y))^2 ) / (length(y)-1)

sxY = sqrt(s2x/n1 + s2y/n2)

TStat = (mean(x) - mean(y)) / sxY

print(TStat)

df = ((s2x/n1 + s2y/n2)^2) / (((s2x/n1)^2) / (n1-1) + ((s2y/n2)^2) / (n2-1))

print(df)

tCriticalValue = qt(1-alpha, df, lower.tail = TRUE)

pVal = pt(TStat, df = df)

print(pVal)

* + - * Výstup:

[1] -2.796436

[1] 37.7662

[1] 0.004040819

* Spočtené hodnoty jsou: t = -2.796436, df = 37.7662 a p-value = 0.004040819.
* Stejně jako dříve jsou hodnoty, které jsme spočetli, stejné, až na menší odchylku.

# Praktické využití t-testů:

## Pro ilustraci praktického využítí t-testů použijeme algoritmu quick sort implementovaného výpočetním systémem R.

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

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

### Ověřte si rychlost svého počítače pomocí kódu

Příkazy R dle instrukcí:

sequenceLength = 2500000;

x = runif(sequenceLength, 0, 100)

print(system.time(sort(x)))

#### Výstup:

user system elapsed

0.434 0.00 0.434

#### Po nastavení sequenceLength na hodnotu 2.5 milionu, jsme dostali čas 0.434, což je mezi hodnotami **0.25 – 0.75**, čehož jsme chtěli dosáhnout. S toutu velikostí bude dále testovat.

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

Příkazy R dle instrukcí:

sampleSize = 40

sampleSize2 = 35

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];

}

meanSingleton = mean(time1)

meanSedgewick = mean(time2)

#### Výstup:

> meanSingleton

[1] 0. 62838

> meanSedgewick

[1] 0. 89932

#### Dokumentace uvádí k metodám funkce sort: *Method "shell" uses Shellsort (an O(n^{4/3}) variant from Sedgewick (1986))…*

*Method "quick" uses Singleton (1969)'s implementation of Hoare's Quicksort method…*

**Nejde tedy o porovnání dvou variant quick sortu**, jak je uvedeno v zadání. Výstup naznačuje, že Shellsort je v tomto případě **o cca 40% pomalejší** než Quicksort. Toto je v souladu s pracovní hypotézou, že **Quicksort je pro velké množiny rychlejší**.

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

Příkazy R dle instrukcí:

alpha = K / 100

t.test(time1, time2, paired = TRUE, var.equal = FALSE, conf.level = 1-alpha, alternative = "less")

#### Výstup:

Paired t-test

data: time1 and time2

t = -32.3444, df = 387.643, p-value < 2.2e-16

alternative hypothesis: true difference in means is not equal to 0

94 percent confidence interval:

-0.2867411 -0.2551389

sample estimates:

mean of x mean of y

0.62838 0.89932

Z výstupu plyne, že **střední hodnota časů řazení Singletonova quicksortu je menší než Sedgewickova**, v souladu s hypotézou.

### Popište přesně jak a proč jste zvolili nulovou hypotézu H0 a alternativu HA.

#### Vybrali jsme H0: muSingleton = muSedgewick , a alternativu Ha:  muSingleton < muSedgewick . Nula do konfidenčního intervalu očividně nepatří, **H0 je možné zamítnout**. Tím získáváme **silný výsledek**, kde pravděpodobnost chybného zamítnutí hypotézy je v našem případě 6%. S 94% pravděpodobností věříme, že střední hodnoty doby setřídění pole pomocí Singletona je menší než střední hodnoty doby seřazení pomocí Sedgewicka, proto jsme vybrali „menší“ alternativu.

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

#### Použili jsme **párový t-test**, neboť se hodí pro porovnání **rozdílu středních hodnot** rozdělení výsledků dvou algoritmů nad **stejnými daty.**

## 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.

Příkazy R dle instrukcí:

#### time3 = time4 = numeric(sampleSize); # Declare an array

#### for(i in 1:sampleSize){

#### x = runif(sequenceLength, 0, 100); # Generate the sequence to be sorted

#### time3[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

#### time4[i] = system.time(x2 <- sort(x, method = "shell"), gcFirst = TRUE)[1];

#### }

#### meanSingleton2 = mean(time3)

#### meanSedgewick2 = mean(time4)

#### Výstup:

> meanSingleton2

[1] 0.641305

> meanSedgewick2

[1] 0. 7398

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

Příkazy R dle instrukcí:

alpha = K / 100

t.test(sep\_time1, sep\_time2, paired = FALSE, var.equal = FALSE, conf.level = 1-alpha, alternative = "less")

#### Výstup:

Welch Two Sample t-test

data: time1 and time2

t = -32.3444, df = 387.643, p-value < 2.2e-16

alternative hypothesis: true difference in means is not equal to 0

94 percent confidence interval:

-0.2867411 -0.2551389

sample estimates:

mean of x mean of y

0.62838 0.89932

### Popište přesně jak a proč jste zvolili nulovou hypotézu H0 a alternativu HA.

#### Vybrali jsme H0: muSingleton = muSedgewick , a alternativu Ha:  muSingleton < muSedgewick . Nula do konfidenčního intervalu očividně nepatří, čili **můžeme H0 zamítnout**. Tím získáváme **silný výsledek**, kde pravděpodobnost chybného zamítnutí hypotézy je v našem případě 4%. S 96% pravděpodobností věříme, že střední hodnoty doby setřídění pole pomocí Singletona je menší než střední hodnoty doby seřazení pomocí Sedgewicka, proto jsme vybrali „menší“ alternativu.

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

#### Použili jsme **nepárový t-test**. Od předchozího případu se liší tím, že dva algoritmy řadí **data, která jsou zcela nezávislá** a výběry jsou různě velké.

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

|  |  |  |
| --- | --- | --- |
|  | Společný vzorek (střední hodnota) | Oddělené vzorky (střední hodnota) |
| **Sedgewick** | 0. 89932 | 0. 7398 |
| **Singleton** | 0. 62838 | 0. 641305 |

V obou experimentech dosahoval **Singletonův Quicksort lepší efektivity** nežli Sedgewickův shellsort, který se pro velká numerická pole ukázal jako nevhodný.