Cvičení z NSTP097
3. 1. 2011

Jednovýběrové a párové testy

Úvod. Následující úlohy řešte samostatně. Svůj zdrojový kód si pište do speciálního skriptu, ať se k němu můžete později vrátit.
Touto barvou budou psány vysvětlení k některým příkazům z R.
Touto barvou budou psány věci, nad kterými byste se měli zamyslet. Porozumění těmto otázkám se může hodit ke zkoušce...
Touto barvou budou psány úkolky, na které se může cvičící ptát. Odpovědi si tedy někam značte.

Doporučený postup práce (*pro ty, kteří si s R-kem stále ještě netykají*)

1. Spusťte program R a nastavte si pracovní adresář, který jste si vytvořili na minulých cvičeních, např. H:/NSTP097.

2. Pomocí File/New script... si založte nový zdrojový soubor a uložte si jej např. jako cviceni3.R. Do tohoto souboru si pište příkazy, které se Vám osvědčily pro řešení zadaných úkolů. Můžete si sem i psát důležité výstupy, které doporučuji „zakomentovat" pomocí znaku # na začátku řádku. Příkazy z tohoto souboru můžete do příkazové konzole R-ka posílat pomocí klávesové zkratky Ctrl+R.

Poznámka k nápovědě: Bohužel v počítačové učebně pod operačním systémem Windows nefunguje standardní nápověda, kdy člověk napíše do příkazového řádku přímo ?plot nebo help(plot) . Jde využít html nápověda, která se vyvolá pomocí help.start() , anebo se k ní lze proklikat pomocí menu Help/... Ta však není příliš šikovná, pokud člověk zná přesný název funkce. Jde však použít úkrok stranou. Pomocí programu putty (bude asi ve skupině Internet) se přihlásíte k serveru artax.karlin.mff.cuni.cz. Zde zadáte svůj login a heslo a po úspěšném přihlášení se ocitnete na svém unixovém účtu na serveru artax. Program R spustíte jednoduše tím, že napíšete do konzole R a dáte Enter. Na takto spuštěné instalaci programu R by měla základní nápověda fungovat již standardně, tj. nápovědu k příkazu plot získáte jako ?plot nebo ekvivalentně help(plot).

Jednovýběrové testy — úvod

Uvažujme náhodný výběr $X_1, \ldots, X_n$ z rozdělení $\mathsf{N}(0.5, 2)$ o rozsahu $n = 60$. Vygenerujte v R jeden takový výběr příkazy

```
nobs = 60
x = rnorm(nobs,mean=0.5,sd=sqrt(2))
```

Jednovýběrový Kolmogorovovův-Smirnovův test.

Provedeme jednovýběrový Kolmogorovovův-Smirnovův test hypotézy $H_0 : X_i \sim \mathsf{N}(0.5, 2)$ proti alternativě, že X_i mají libovolné jiné rozdělení. V R se takový KS test na výběru `x` provede příkazem

```
ks.test(x,y="pnorm",mean=0.5,sd=sqrt(2))
```

funkce `ks.test` provádí Kolmogorovovův-Smirnovův test, `pnorm` specifikuje distribuční funkci normálního rozdělení a `mean=0.5,sd=sqrt(2)` zadává parametry tohoto rozdělení. Je samozřejmě možné pomocí `ks.test` testovat i jiná rozdělení — např. `pgamma` aj.

Prozkoumejte výstup z funkce `ks.test`:

1. Jaká je hodnota testové statistiky a p-hodnoty? Zamítáme nulovou hypotézu na hladině 0.05?

Ujistěte se, že víte, co udává p-hodnota.
Připomeňte si, na čem je založen Kolmogorovovův-Smirnovův test — co porovnává a jak?

Nakreslíme si obrázek empirické distribuční funkce spolu s distribuční funkcí za hypotézy:

```
od = min(x)*0.9
do = max(x)*1.1
plot(ecdf(x),xlim=c(od,do))
curve(pnorm(x,mean=0.5,sd=sqrt(2)),od,do,col="blue",add=T)
```

Do parametrů `od` a `do` nastavujeme rozsah osy x pro kreslení obrázku, `ecdf(x)` počítá empirickou distribuční funkci (měli jsme minule) a poslední příkaz přidá do obrázku teoretickou distribuční funkci normálního rozdělení $\mathsf{N}(0.5, 2)$.

Dá se z obrázku okem odhadnout hodnota testové statistiky KS testu?

Nyní na ten samý výběr `x` proveďte postupně test hypotézy $H_0 : X_i \sim \mathsf{N}(\mu, 2)$ pro $\mu = 0.60, 0.65, 0.70, \ldots, 1.00$. Příslušné příkazy můžete zadávat do `R` postupně nebo si je zavolat najednou pomocí

```
posl.mu=seq(0.6,1.5,by=0.05)
for (mu in  posl.mu){ print(mu)
                      print(ks.test(x,y="pnorm",mean=mu,sd=sqrt(2)) )
                    }
```

Do `posl.mu` si uložíme posloupnost 0.6, 0.65, …, 1 a pak pro všechny její prvky pomocí `for` cyklu provedeme `ks.test`.

Jak se mění výsledek testu?

Pokračujte v oddalování střední hodnoty hypotetického rozdělení od střední hodnoty skutečného rozdělení dat po stejných krůčcích, dokud nedojde k zamítnutí hypotézy. Pro hodnotu, pro kterou jste zamítli nulovou hypotézu (označme ji např. `mu.zam`), si nakreslete obrázek empirické distribuční funkce dat a distribuční funkce za platnosti hypotézy.

```
plot(ecdf(x),xlim=c(od,do))
curve(pnorm(x,mean=mu.zam,sd=sqrt(2)),od,do,col="blue",add=T)
```

Vidíte rozdíl oproti prvnímu obrázku, co jste si dnes kreslili?

2. Pro jaké μ došlo k zamítnutí nulové hypotézy na hladině 0.05 (resp. 0.01)?

Jednovýběrový t-test.

Zopakujte stejné zadání s t-testem: Nejdříve proveďte t-test hypotézy $H_0 : \mathsf{E}X_i = 0.5$ proti alternativě $H_1 : \mathsf{E}X_i \neq 0.5$ na výběru `x` příkazem
`t.test(x,mu=0.5)`
a prozkoumejte výstup z funkce `t.test`.

3. Čemu je rovna testová statistika a p-hodnota? Zamítáme nulovou hypotézu na hladině 0.05?

Jaké je rozdělení testové statistiky za hypotézy? Uměli byste spočítat p-hodnotu „ručně“ pouze z hodnoty testové statistiky?

Zkoumejte výsledky t-testu hypotéz $H_0 : \mathsf{E}X_i = \mu$ pro $\mu = 0.6,\ 0.65,\ 0.7,\ 0.75, \ldots$, dokud nedojde k zamítnutí H_0.

```
posl.mu=seq(0.6,1.5,by=0.05)
for (mu in  posl.mu){ print(mu)
                      print(t.test(x,mu=mu))
                     }
```

4. Došlo k zamítnutí dříve nebo později než u KS testu?

Který test byste doporučili? A je vůbec toto porovnávání testů férové?

Jednovýběrové testy — simulace, hladina, síla, p-hodnota

Nyní budeme simulovat hladinu a sílu jednovýběrových testů. Připomeňte si, co je to hladina a síla testu.

Vyrobíme si jednoduchou funkci, která provede test na data `x` a vrátí pouze p-hodnotu.

```
vem.ph = function(x,test,...)
  {
    test(x,...)$p.value
  }
```

Proměnná `test` obsahuje název funkce v `R`-ku, která provádí daný test. Tři tečky ... naznačují všechny ostatní parametry testu. Tyto parametry se mohou lišit pro různé testy.

Vyzkoušejte si tuto novou funkci na původní data `x` s K-S testem a t-testem (porovnáním s předchozími výstupy ověřte, že to skutečně „funguje“):
`vem.ph(x,ks.test,y="pnorm",mean=0.5,sd=sqrt(2))`
`vem.ph(x,t.test,mu=0.5)`

Teď vygenerujeme 1000 výběrů o rozsahu 60 z rozdělení $\mathsf{N}(0.5, 2)$:
`nvyb = 1000`
`x.vyb = matrix(rnorm(nobs*nvyb,0.5,sqrt(2)),nrow=nobs,ncol=nvyb)`

Matice `x.vyb` má v `nvyb` sloupců a v každém sloupci je náhodný výběr z $\mathsf{N}(0.5, 2)$ o rozsahu `nobs`.

Na každý z výběrů provedeme K-S test hypotézy $H_0 : X_i \sim \mathsf{N}(0.5, 2)$ a získáme jeho p-hodnotu:
`ks.ph = apply(x.vyb,2,vem.ph,ks.test,y="pnorm",mean=0.50,sd=sqrt(2))`

Aplikovali jsme funkci `vem.ph` s testem `ks.test` na sloupce matice `x.vyb`. Volba 2 v `apply` říká, že chceme funkce aplikovat na sloupce, `y="pnorm",mean=0.50,sd=sqrt(2)` specifikuje parametry funkce `ks.test`.

Vektor `ks.ph` obsahuje p-hodnoty pro `nvyb` (tj. 1 000) výběrů z rozdělení za platnosti hypotézy. Nakreslete si jejich histogram `hist(ks.ph)` .

Výsledné p-hodnoty jsou náhodné veličiny: Jaké je v tomto případě jejich rozdělení? Váš úsudek si teoreticky zdůvodněte a **ověřte** provedením K-S testu na výběr `ks.ph` 1000 p-hodnot.
`ks.test(ks.ph,y=??)`
za `??` musíte doplnit distribuční funkci rozdělení p-hodnot za nulové hypotézy

Spočtěte, jaký podíl p-hodnot je menších než 0.05
`mean(ks.ph<0.05)`.
5. Poznamenejte si toto číslo a popřemýšlejte, co odhaduje v řeči testování hypotéz. Čemu by mělo být přibližně rovno?

6. Pro následující scénáře si poznačte podíl p-hodnot, které jsou menší než 0.05.

- Generujte výběry z $\mathsf{N}(\mu, 2)$ pro $\mu = 0.7$ a $\mu = 0.9$, testujte stále hypotézu $H_0 : X_i \sim \mathsf{N}(0.5, 2)$ K-S testem.

  ```
  nvyb = 1000
  x.vyb = matrix(rnorm(nobs*nvyb,0.7,sqrt(2)),nrow=nobs,ncol=nvyb)
  ks.ph = apply(x.vyb,2,vem.ph,ks.test,y="pnorm",mean=0.5,sd=sqrt(2))
  hist(ks.ph)
  mean(ks.ph<0.05)
  ```

  Změnilo se rozdělení p-hodnot? Mění se počet p-hodnot menších než 0.05? Co to znamená? Co nyní odhaduje podíl p-hodnot menších než 0.05?

- Generujte výběry z $\mathsf{N}(0.5, 2)$ a provádějte t-test hypotézy $H_0 : \mathsf{E}X_i = 0.5$. Interpretujte výsledky.

  ```
  nvyb = 1000
  x.vyb = matrix(rnorm(nobs*nvyb,0.5,sqrt(2)),nrow=nobs,ncol=nvyb)
  t.ph = apply(x.vyb,2,vem.ph,t.test,mu=0.5)
  hist(t.ph)
  mean(t.ph<0.05)
  ```

- Generujte výběry z $\mathsf{N}(\mu, 2)$ pro $\mu = 0.7$ a $\mu = 0.9$, testujte hypotézu $H_0 : \mathsf{E}X_i = 0.5$ t-testem. Interpretujte výsledky.
  Porovnejte podíl p-hodnot menších než 0.05 v případě K-S testu a t-testu. Co nám to říká?

Párové testy — simulace

Nyní uvažujme náhodný vektor $\binom{X}{Y}$, kde $X \sim \mathsf{N}(0.5, 2)$ a $Y = X + \varepsilon$, kde $\varepsilon \sim \mathsf{N}(0, 1)$ je nezávislé na X.

7. Spočítejte (teoreticky) $\mathsf{Cov}\,(X, Y)$ a $\mathsf{Cor}\,(X, Y)$.

Jaké je sdružené rozdělení $\binom{X}{Y}$?

Vygenerujme jeden výběr dvojic $\binom{X_1}{Y_1}, \ldots, \binom{X_n}{Y_n}$ pro $n = 60$:

```
nobs = 60
x = rnorm(nobs,mean=0.5,sd=sqrt(2))
y=x+rnorm(nobs,mean=0,sd=1)
```

8. Spočítejte výběrový korelační koeficient mezi X_i a Y_i
`cor(x,y)`
a porovnejte jej se skutečnou korelací $\mathsf{Cor}\,(X, Y)$.

Spočítejte průměry X_i a Y_i
`mean(x); mean(y)`

Proveďte párový t-test hypotézy $H_0 : \mathsf{E}X_i = \mathsf{E}Y_i$ proti alternativě $H_1 : \mathsf{E}X_i \neq \mathsf{E}Y_i$:
`t.test(x,y,paired=T)`
a prozkoumejte jeho výstup

9. Identifikujte všechny komponenty výstupu, zejména testovou statistiku a p-hodnotu. Zamítáme nulovou hypotézu na hladině 0.05?

Proč provádíme párový t-test a ne dvouvýběrový t-test?

Nyní generujte znovu `y` při pevném `x`, přičemž po krocích velkých 0.05 postupně zvětšujte střední hodnotu $\mathsf{E}Y_i$, dokud nedojde k zamítnutí H_0. (Tzn. měníte střední hodnotu ε.)

```
x =rnorm(nobs,mean=0.5,sd=sqrt(2))
posl.mu=seq(0,0.5,by=0.05)
eps = rnorm(nobs,mean=0,sd=1)
for (mu in  posl.mu){  y = x + eps + mu
                       print(mu)
                       print(t.test(x,y,paired=T))
                    }
```

10. Jak velké μ stačilo k zamítnutí nulové hypotézy?

Nyní znovu provedeme simulace p-hodnot, ale musíme upravit funkci `vem.ph` pro výpočet p-hodnot na párový test. Data budeme ukládat do matice o `nvyb=1000` sloupcích tak, aby prvních 60 prvků každého sloupce obsahovalo X_i a druhých 60 prvků odpovídající Y_i. Funkce pro výpočet p-hodnot bude

```
vem.ph.2 = function(data,test,...)
  {
```

```
    test(x=data[1:nobs],y=data[-c(1:nobs)],...)$p.value
  }
```

Příkaz `data[1:nobs]` bere prvních 1 až `nobs` prvků z `data`, příkaz `data[-c(1:nobs)]` bere zbývající prvky, tj. posledních `nobs` prvků z `data`.

Vyzkoušejte si funkci `vem.ph.2`
`vem.ph.2(c(x,y),t.test,paired=T)`
Zkontrolujte, že jste dostali stejnou p-hodnotu jako z
`t.test(x,y,paired=T)`.

Nyní dáme dohromady výběry pro $X \sim \mathsf{N}(0.5, 2)$ a $Y = X + \varepsilon$, kde $\varepsilon \sim \mathsf{N}(0, 1)$,

```
nvyb = 1000
x.vyb = matrix(rnorm(nobs*nvyb,0.5,sqrt(2)),nrow=nobs,ncol=nvyb)
y.vyb = x.vyb+matrix(rnorm(nobs*nvyb,mean=0,sd=1),nrow=nobs,ncol=nvyb)
oba.vyb = rbind(x.vyb,y.vyb)
```

Příkaz `rbind` spojí dvě matice pod sebe.
Vypočteme 1000 p-hodnot
`t.ph = apply(oba.vyb,2,vem.ph.2,t.test,paired=T)`
Nakreslete si jejich histogram, otestujeme jejich rozdělení, a spočtěte, kolik jich bylo pod 0.05.

V dalším kroku opakujeme simulace se změnou střední hodnoty veličiny ε: místo 0 vezměte 0.1, 0.2, 0.3, a 0.4. Zadávejte postupně nebo pomocí

```
# Nastavim si okno, aby se do nej zakreslili 4 grafy jako 2x2
 par(mfrow=c(2,2))
 mu.seq=c(0.1,0.2,0.3,0.4)
 for(mu in mu.seq){
# vytvorim si vybery y, vybery x jiz mam z drivejska
    y.vyb = x.vyb + matrix(rnorm(nobs*nvyb,mean=mu,sd=1),nrow=nobs,ncol=nvyb)
# vybery spojim do matice, tak ze v kazdem sloupci je nejdrive vyber x
# a pak vyber y
    oba.vyb = rbind(x.vyb,y.vyb)
# v kazdem sloupci udelam parovy t-test a vezmu si z nej pouze p-hodnotu
    t.ph = apply(oba.vyb,2,vem.ph.2,t.test,paired=T)
# nakreslim histogram p-hodnot
    hist(t.ph,main=paste("mu=",mu))
# vytisknu silu testu
    print(paste("mu=",mu,"podil=",mean(t.ph<0.05)))
# otestuji, zda se daji p-hodnoty povazovat za vyber z rovnomerneho rozdeleni
# na (0,1)
    print(ks.test(x=t.ph,y="punif"))
}
```

11. Poznačte si, jak se mění vývoj podílu p-hodnot menších než 0.05 a promyslete si, zda to odpovídá tomu, co byste očekávali.

Párové testy — data „Pneu“

Na minulém cvičení jsme pracovali s datovým souborem `pneu`. Připomeňme, že obsahuje měření životnosti pneumatik (v tisících km jízdy) dvěma metodami: měření pomocí úbytku hmotnosti pneumatiky (veličina `met.v`) a měření pomocí úbytku hloubky dezénu (veličina `met.d`).

Z webové stránky `www.karlin.mff.cuni.cz/~omelka/Vyuka_stp097.php` si stáhněte soubor `cviceni3.RData`. Otevřete si program `R` a pomocí `File/Load_Workspace...` si načtěte do `R` tento soubor. Natáhli jste si tak znovu zmíněná data a navíc jednu další funkci. Nezapomeňte si data zpřístupnit pomocí `attach(pneu)`.

Spočítejte párový t-test srovnávající obě metody měření.

```
t.test(met.v, met.d, paired=T)
```

12. Najděte ve výstupu testovou statistiku a p-hodnotu. Promyslete si, co je nulová hypotéza a zda ji zamítáme. Zamítnutí či nezamítnutí nulové hypotézy interpretujte.

Spočítejte **kritickou hodnotu** pro testovou statistiku. Proveďte **vlastní výpočet** p-hodnoty z testové statistiky.

Návod: α-kvantil t-rozdělení s k stupni volnosti dostaneme jako `qt(alfa,df=k)`, distribuční funkci v bodě x jako `pt(x,df=k)`.

Proveďte **znaménkový test**. Máte k dispozici funkci `sign.test`, kterou jste si natáhli s daty (původně pochází z knihovny `BSDA`). Zavolejte
`sign.test(met.v, met.d)` .

13. Jakou nulovou hypotézu testuje tento test? Zamítá ji?

Jak se spočítá testová statistika? Spočítejte si sami v `R` tuto testovou statistiku.

Jaké je teoretické rozdělení testové statistiky za nulové hypotézy?