METODY MONTE CARLO V EKONOMETRII.

MAKROEKONOMICKÁ ÚLOHA VLÁDY

V ekonometrii se někdy objeví problémy, které nelze řešit analytickými postupy, nebo je to přinejmenším velmi obtížné. V tom případě je vhodné použít simulační postupy.

simulační postupy	induktivní specifikace = odvozují obecná pravidla na základě konkrétních případů	výběrová informace
analytické postupy	deduktivní specifikace = vyvozují závěry pro konkrétní případ na základě obecných pravidel	apriorní informace

Simulací se rozumí numerická technika, pomocí které experimentujeme s odhadnutým modelem, abychom prozkoumali jeho vlastnosti. Přijímáme přitom určité předpoklady o hodnotách parametrů a proměnných či o rozdělení náhodných složek. Závěry mají pouze pravděpodobnostní charakter, jejich přesnost ale roste s počtem pokusů.

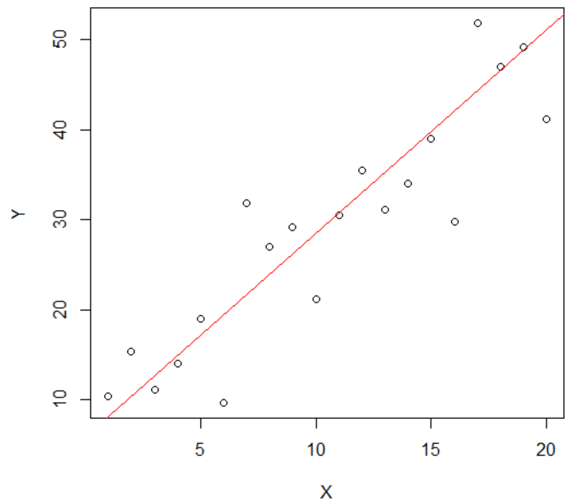
ex post (historická)	Experimentujeme s různými specifikacemi modelu a pracujeme přitom s již známými pozorováními použitými k odhadu tohoto modelu. Porovnáním skutečných a nasimulovaných hodnot můžeme posoudit vhodnost zvolené specifikace modelu.
ex ante (projekce)	Generujeme budoucí hodnoty či obecně hodnoty mimo interval pozorování. Lze tak dělat např. pseudopředpovědi ex post, a posoudit tak predikční schopnost modelu. Lze použít i při restrospektivě pro experimenty v intervalu před počátkem období pozorování za účelem ověření dynamické stability modelu.
deterministická	Negenerujeme náhodné veličiny
stochastická	Generujeme náhodné veličiny, např. náhodné složky modelu, z určitého pravděpodobnostního rozdělení. Pracujeme tedy s umělými daty místo skutečných údajů. Stochastická simulace se nazývá také simulace Monte Carlo. Její speciální formou je bootstrap.

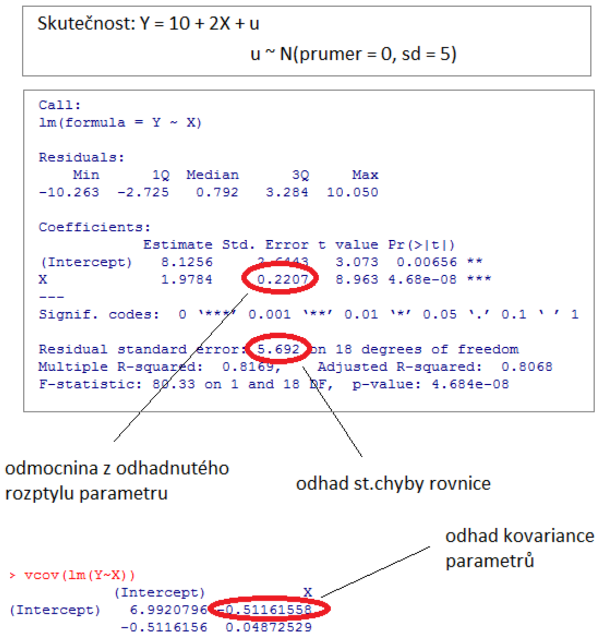
MONTE CARLO

Monte Carlo má původ ve 40. letech 20. století a její název je odvozen od kasina v Monaku. Jde o statistický experiment, v rámci kterého vygenerujeme velké množství náhodných čísel a pak z nich vyvozujeme určité závěry, tzn. mnohonásobně opakujeme tentýž proces. Monte Carlo je vlastně řešením určité úlohy pomocí statistického experimentu. Simulace se vyvinula právě z metody Monte Carlo a věnuje se studiu rozsáhlých dynamických systémů (Monte Carlo neobsahuje dynamiku).

V ekonometrii většinou generujeme aditivní náhodné složky modelu, hodnoty parametrů či proměnných, a to z určitého pravděpodobnostního rozdělení.

- ➔ u náhodných složek většinou předpokládáme, že mají normální rozdělení s nulovou střední hodnotou a směrodatnou odchylkou rovnou odhadnuté standardní chybě dané rovnice
- ➔ u parametrů většinou předpokládáme simultánní normální rozdělení. Pro zanedbatelnou kovarianci můžeme předpokládat, že pochází z normálního rozdělení s průměrem rovným odhadu parametru a směrodatnou odchylkou rovnou odhadnuté standardní chybě parametru = odmocnině z odhadnutého rozptylu parametru. Kovarianci zjistíme z kovarianční matice odhadnutých parametrů.





Vhodnost odhadové funkce posuzujeme podle různých kritérií. Označme β skutečnou hodnotu parametru a $\bar{b}$ průměrný odhad tohoto parametru. Udělali jsme *N* výběrů a z každého získali odhad parametru b_p.

- ➔ vychýlení odhadové funkce: $\bar{b} - \beta$
- ➔ rozptyl odhadové funkce: $\frac{1}{N}\sum_{p=1}^{N}(b_p - \bar{b})^2$
- ➔ střední kvadratická chyba: $\frac{1}{N}\sum_{p=1}^{N}(b_p - \beta)^2 = \frac{1}{N}\sum_{p=1}^{N}(b_p - \bar{b})^2 + (\bar{b} - \beta)^2$

Střední kvadratická chyba se rovná součtu rozptylu a čtverce vychýlení, takže pro nevychýlenou odhadovou funkci se rovná pouze rozptylu. Střední kvadratická chyba má smysl pro případ, kdy jsou odhady normálně rozděleny (jinak je lepší použít např. medián).

Využití simulace:

- **prozkoumání asymptotických vlastností různých metod odhadů**
- **testování a zpřesňování specifikace modelu**
- **ověření predikční schopnosti modelu**. Lze pracovat s předpovědí ex ante i ex post. V případě předpovědi ex ante na základě odhadnutých parametrů a rozptylu (či kovarianční matice náhodných složek v případě MSR) opakovaně spočítáme ex ante předpověď a porovnáme ji s **deterministickou podmíněnou předpovědí**. V případě předpovědi ex post porovnáváme simulační předpověď endogenní proměnné se **skutečně realizovanou hodnotou**. Většinou pracujeme se střední kvadratickou chybou nebo její odmocninou (RMSE). Neparametrickou mírou vhodnosti modelu k predikci je Theilův modifikovaný koeficient nesouladu (viz otázka 3C Prognózování).
- **hospodářská politika**: zjištění vlivu nastavení řídících proměnných na hodnoty cílových proměnných (scénářová analýza). Většinou pracujeme s dynamickými MSR. Podstatou je řešení daného tvaru MSR pro různé hodnoty řídících či cílových proměnných, náhodných složek, funkční tvary modelu apod. **Variantní analýza** slouží k posouzení vlivů různých variant strategie řízení na úroveň sledovaných cílů. **Cílová analýza** vyhodnocuje různé možné způsoby, jak dosáhnout zvolených hodnot cílových proměnných.

Příklady použití Monte Carlo simulace

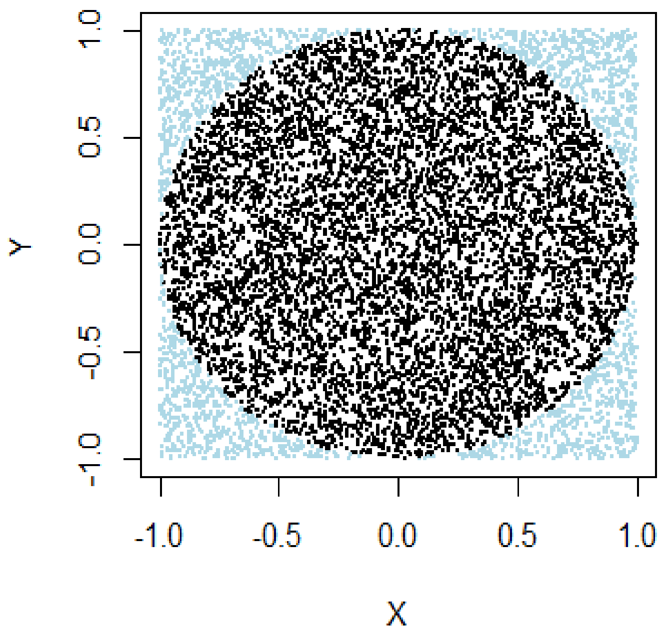
1. Odhad Ludolfova čísla (úvod do simulace Monte Carlo)
2. Odhad integrálu
3. Simulace s MSR
4. Zkoumání vlastností odhadových funkcí
5. Síla a validita testů hypotéz
6. Odhad směrodatné chyby průměru pomocí bootstrapu
7. Bootstrap v regresi

PŘÍKLAD 1: ODHAD LUDOLFOVA ČÍSLA

Jedním z nejjednodušších příkladů použití Monte Carlo simulace je odhad čísla π. Vygenerujeme velké množství bodů ve čtverci o rozměru 2r x 2r, kde r = 1. Uvažujme kruh o poloměru 1 vepsaný tomuto čtverci. Podíl obsahu kruhu a obsahu čtverce je roven $(\pi r^2)/(4r^2) = \pi/4$. Vygenerujeme 10 000 bodů ležících v tomto čtverci. Čtyřnásobek podílu počtu pouze těch bodů, které leží v kruhu, na všech vygenerovaných bodech je odhadem čísla π.

```
r = 1
k = 0
plot(0~0, xlab = "X", ylab = "Y",pch=16, cex = 0.1, ylim = c(-1,1), xlim = c(-1,1))
for (i in 1:10000)
{       x = runif(1,-1,1) #vygenerujeme nahodne cislo z interval -1,1 = souradnice x
        y = runif(1,-1,1) #vygenerujeme nahodne cislo z interval -1,1 = souradnice y
        points(y~x, pch = 16, cex = 0.1, col = ifelse(sqrt(x*x + y*y) <= r, "black", "lightblue"))
        #spocitame Pythagorovou vetou jeho vzdalenost od stredu
        #pokud toto cislo lezi v kruhu, pak navysime k o 1:
        if (sqrt(x*x + y*y) <= r)
                        k = k+1 }
4*k/10000 #odhad cisla pi
```

Výsledek mi při jednom konkrétním experimentu vyšel 3,134. Skutečná hodnota je 3,141



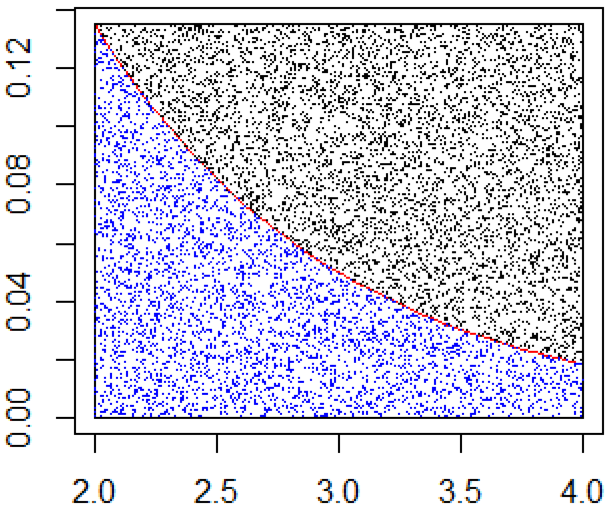
PŘÍKLAD 2: ODHAD INTEGRÁLU

Odhadneme hodnotu integrálu $\int_2^4 e^{-x} dx$. Budeme postupovat podobně jako v předchozím případě. Vygenerujeme náhodnou souřadnici *x* z rovnoměrného rozdělení na intervalu (2,4). Pak vygenerujeme náhodnou souřadnici *y* z rovnoměrného rozdělení na intervalu $(0,e^{-2})$. Celkem těchto souřadnic vygenerujeme 10 000. Všechny tyto body leží tedy v obdélníku o obsahu 2 krát e^{-2}. Zjistíme, pro jaký podíl bodů platí, že e^{-x} je menší než *y*. Tento podíl vynásobený obsahem obdélníka je odhadem integrálu.

Skrip napsal M.Basta z KTSP:

```
hh = exp(-2) # y-ova souradnice horni hranice obdelnika
S = (4 - 2) * hh # plocha obdelnika
# vykreslime si pomocny obrazek
xseq = seq(2, 4, by = 0.05)
plot(xseq, exp(-xseq), type = "l", col = "red", ylim = c(0, hh))
rect(2, 0, 4, hh)
# nahodne vygenerujeme n bodu uvnitr obdelniku
n = 10000
x = runif(n, min = 2, max = 4)
y = runif(n, min = 0, max = hh)
# vygenerovane body zakreslime
points(x, y, pch = ".", col = ifelse(y < exp(-x), "blue", "black"))
odhP = sum(y < exp(-x))/n # relativni cetnost zasahu
(odhInt = S * odhP) # odhad integralu
# skutecna hodnota integralu
exp(-2) - exp(-4)
```

Výsledek mi při jednom konkrétním experimentu vyšel 0,115. Skutečná hodnota je 0,117.



PŘÍKLAD 3: SIMULACE S MSR

Uvažujme následující soustavu rovnic (model multiplikátor – akcelerátor):

$$C_t = \alpha_1 + \alpha_2 Y_{t-1} + u_{1t}$$
$$I_t = \beta_1 + \beta_2 (Y_{t-1} - Y_{t-2}) + u_{2t}$$
$$Y_t = C_t + I_t + G_t$$

kde *C* je konečná spotřeba, *I* jsou hrubé investice, *Y* je HDP, *G* jsou veřejné výdaje a *u* jsou náhodné složky. Veřejné výdaje jsou exogenní proměnnou, zbylé tři jsou endogenními proměnnými. Zajímá-li nás řešení pro proměnnou Y_t, dosadíme první dvě rovnice (bez náhodných složek) do poslední rovnice a dostaneme tzv. fundamentální dynamickou rovnici, kde a_1, a_2, b_1, b_2 jsou známé odhady strukturních parametrů:

$$Y_t - (a_2 + b_2)Y_{t-1} + b_2 Y_{t-2} = (a_1 + b_1) + G_t$$

Pokud děláme stochastickou Monte Carlo simulaci, musíme do modelu přidat vliv náhody. Budeme tedy potřebovat:

- e_{1t}, e_{2t} – rezidua, o kterých předpokládáme, že mají normální rozdělení s nulovou střední hodnotou a rozptyl rovný čtverci odhadnuté standardní chyby příslušné rovnice
- ε_{11}, ε_{12}, ε_{21}, ε_{22} – náhodné chyby odhadnutých strukturních parametrů. Jde o simultánně normálně rozdělené náhodné veličiny. Pokud je jejich kovariance zanedbatelná, můžeme je aproximovat normálním rozdělením s nulovou střední hodnotou a rozptylem rovným čtverci odhadnuté standardní chyby příslušeného parametru.

Po zahrnutí těchto náhodných veličin lze model přepsat jako:

$$C_t = (a_1 + \varepsilon_{11}) + (a_2 + \varepsilon_{12})Y_{t-1} + e_{1t}$$
$$I_t = (b_1 + \varepsilon_{21}) + (b_2 + \varepsilon_{22})(Y_{t-1} - Y_{t-2}) + e_{2t}$$
$$Y_t = C_t + I_t + G_t$$

Fundamentální dynamická rovnice pro Y_t pak je:

$$Y_t - [(a_2 + \varepsilon_{12}) + (b_2 + \varepsilon_{22})]Y_{t-1} + (b_2 + \varepsilon_{22})Y_{t-2} - e_{1t} - e_{2t} = (a_1 + \varepsilon_{11}) + (b_1 + \varepsilon_{21}) + G_t$$

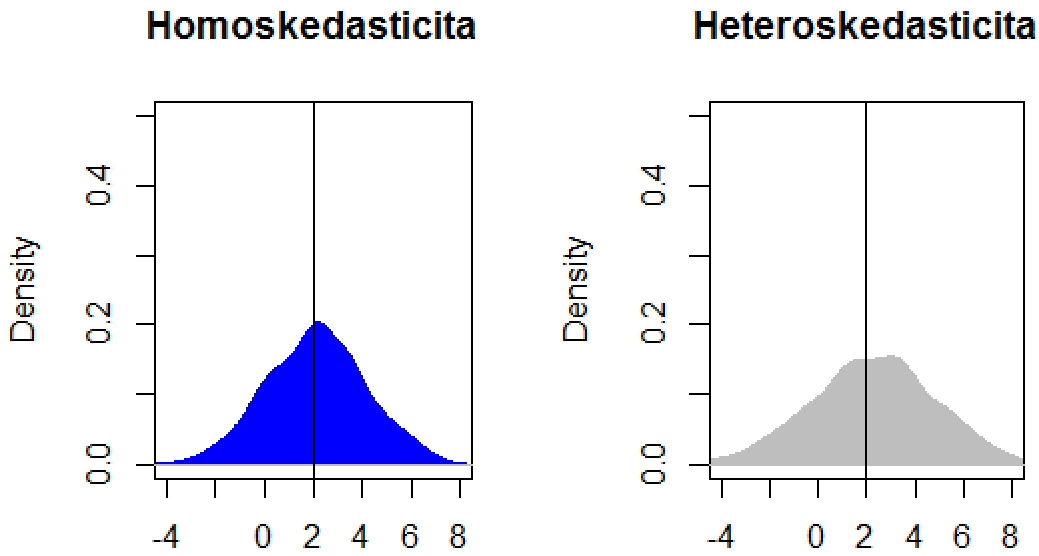
Následně můžeme generovat trajektorie hodnot Y_t pro různé hodnoty ε_{11}, ε_{12}, ε_{21}, ε_{22}, přičemž tyto hodnoty považujeme za dané a hodnoty reziduí generujeme náhodně z daného pravděpodobnostního rozdělení.

Příklad 4: Zkoumání vlastností odhadových funkcí

Vygenerujeme náhodných 100 hodnot nezávislých proměnných X z rovnoměrného rozdělení (1, 20). Budeme uvažovat dva případy. V obou případech bude vysvětlovaná proměnná Y generována vztahem $Y = \alpha + \beta X + u$, konkrétně $Y = 50 + 2X + u$. Rozdíl bude v náhodné složce. V prvním případě bude náhodná složka homoskedastická a bude pocházet z normálního rozdělení s průměrem 0 a směrodatnou odchylkou rovnou průměrné hodnotě X. Ve druhém případě bude náhodná složka heteroskedastická. Její průměr bude roven nule, avšak její rozptyl bude funkcí druhé mocniny X, tzn. $\sigma_i^2 = (\overline{X} \cdot X_i)^2$. Pro každý z případů vygenerujeme 100 hodnot náhodných složek a spočítáme 100 hodnot vysvětlovaných proměnných výše uvedeným vztahem. Pak tento vztah jakoby zapomeneme a zkusíme z těchto hodnot parametry zpětně odhadnout. Odhad parametru β by se měl blížit dvěma. Zaznamenáme jeho hodnotu a také odhad jeho směrodatné chyby, který je rovněž výstupem modelu. Totéž provedeme tisíckrát.

Tak získáme 1000 odhadnutých hodnot pro případ homoskedasticity a dalších 1000 pro případ heteroskedasticity. Vygenerujeme graf hustoty odhadové funkce parametru β pro oba případy. V obou případech také porovnáme směrodatnou chybu spočítanou z této tisícovky hodnot, která udává variabilitu odhadové funkce, s průměrným odhadem směrodatné chyby z modelu. Tyto hodnoty by se měly rovnat, víme ale, že pro případ heteroskedasticity jsou odhady směrodatné chyby vychýlené, proto se asi rovnat nebudou.

```
X = runif(100,1,20) #vygenerujeme vysvetlovanou promennou
sd = mean(X)
#homoskedasticita
b =rep(NA,1000)
StE = NULL
for (i in 1:1000) #pro kazde z 1000 opakovani…
{ Y = 50 + 2*X + rnorm(100,0,sd*sd) #vygenerujeme nahodnou slozku a spocitame Y
regrese.puvodni=lm(Y~X) #jakoby zapomeneme skutecne parametry a odhadneme je MNC
b[i] = regrese.puvodni$coef[2] #ulozime si odhad parametru beta
StE = c(StE, coef(summary(regrese.puvodni))[, "Std. Error"][2])} #a odhad jeho smerodatne chyby
sd(b) - mean(StE)  # rozdil smerodatne odchylky 1000 odhadnutzch parametru a prumerne sm.chyby
#konkretni vysledek, když jsem to zkousela: -0.027
#heteroskedasticita – tentyz postup s tim rozdilem, ze rozptyl nahodne slozky je funkci X
h =rep(NA,1000)
StE2 = NULL
for (i in 1:1000)
{ u = rep(NA,100)
for(j in 1:100)  #rozptyl je funkci ctvercu vysv.promennych (v R se pri generovani udava sm.odchylka)
        {       u[j] = rnorm(1,0,sd*X[j])}
                Y = 50 + 2*X + u
                regrese=lm(Y~X)
                h[i] = regrese$coef[2]
StE2 = c(StE2, coef(summary(regrese))[, "Std. Error"][2]) }
sd(h) - mean(StE2)
#konkretni vysledek, když jsem to zkousela: 0.25, zda se, ze odhad sm.chyby je vychyleny
#grafy:
par(mfrow = c(1,2))
plot(density(b),xlim = c(-4,8),ylim = c(0,0.5),main = "Homoskedasticita",type = "h", col = "blue")
abline(v=2)
plot(density(h),xlim = c(-4,8),ylim = c(0,0.5),main = "Heteroskedasticita", type = "h", col = "grey")
abline(v=2)
```



Simulací jsme zjistili, že **MNČ poskytuje nevychýlené odhady i v případě heteroskedasticity**, avšak odhad směrodatné chyby je pro případ heteroskedasticity **vychýlený**.

Následně bychom pro případ heteroskedasticity mohli porovnat odhady při aplikaci MNČ (získané výše) a MZNČ. Zjistili bychom pravděpodobně, že variabilita odhadů při použití MZNČ je menší, tedy že MZNČ je vydatnější ve srovnání s MNČ.

Totéž je možné provést i pro případ autokorelace (viz Hušek (2007), str. 327-328). Pro fixní hodnoty vysvětlujících proměnných a různé koeficienty autokorelace bychom opakovaně generovali náhodné složky a dopočítali hodnotu vysvětlované proměnné podle předem specifikovaného vztahu: $Y = 15 + X_1 + 2X_2 + u$. Ten bychom pak zapomněli a parametry zpětně odhadovali pomocí MNČ a MZNČ. Zjistili bychom, že pro vyšší hodnoty koeficientu autokorelace je variabilita odhadů nižší při použití MZNČ ve srovnání s MNČ, což znamená, že odhadová funkce MNČ není vydatná. Také bychom zjistili, že odhad směrodatné chyby parametrů je při použití MNČ vychýlený (podhodnocený).

PŘÍKLAD 5: SÍLA A VALIDITA TESTŮ HYPOTÉZ

Když chceme zjistit **sílu testu**, zkoumáme, jaká je pravděpodobnost, že skutečně zamítneme nulovou hypotézu, pokud neplatí. Síla testu klesá s klesající hladinou významnosti.

Když chceme zjistit **validitu testu**, zkoumáme, jaká je pravděpodobnost, že zamítneme nulovou hypotézu, která přitom platí (tzn. že se dopustíme chyby prvního druhu). Tato pravděpodobnost by se měla rovnat hladině významnosti α. Tedy například při α = 0,05 bychom mohli testovat, zda se průměr výšky v populaci rovná 170 cm. Uděláme 1 000 výběrů a provedeme test hypotézy. Přibližně v 50 výběrech bychom měli hypotézu, že je průměrná výška rovna 170 cm, na dané hladině významnosti zamítnout. Stane se to tehdy, když náhodou vybereme hodně vysokých či naopak hodně nízkých lidí, a to se občas stane.

Pomocí simulace Monte Carlo je možné testovat sílu a validitu testu, což je v ekonometrii velmi důležité. Uvažujme Shapiro-Wilk test normality. Budeme opakovaně generovat výběry různého rozsahu: nejprve z t-rozdělení s 10 stupni volnosti, pak z uniformního rozdělení a nakonec z normálního rozdělení. Zjistíme pro každý výběr p-hodnotu. Pokud bude nižší než 0,05, znamená to, že na 5 % hladině významnosti bychom zamítli nulovou hypotézu o normalitě výběru.

Pro výběry z t-rozdělení a uniformního rozdělení tímto ověříme sílu testu. Pokud je test dostatečně silný, měl by nulovou hypotézu o normalitě zamítnout, jelikož neplatí. Pro výběry z normálního rozdělení tímto ověříme validitu testu. Validní test by měl zamítnout nulovou hypotézu v 5 % případů.

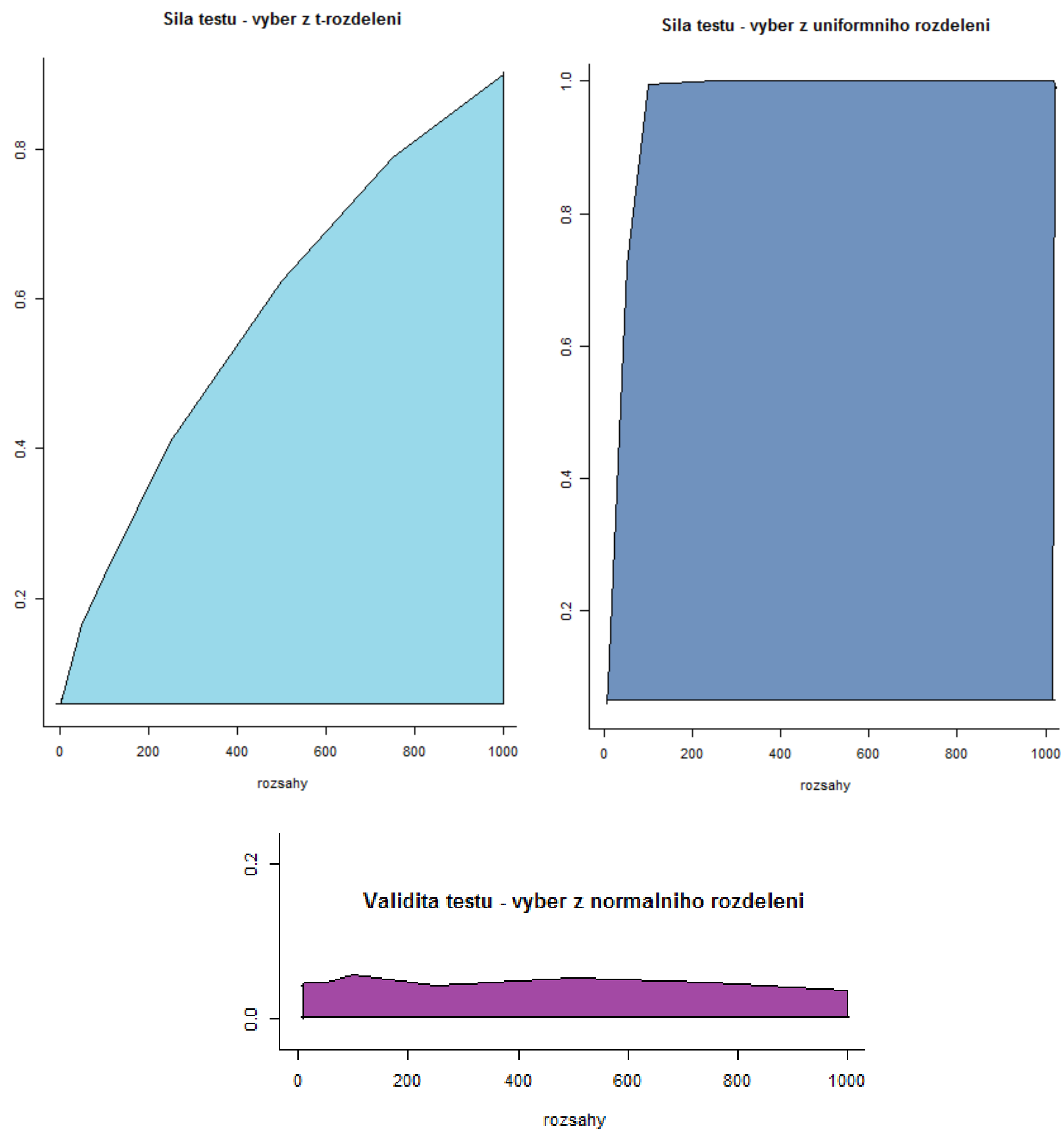
```
Skript – M.Basta (upraveno)
phodnota = function(vyb) {
  return(shapiro.test(vyb)$p.value)}  # funkce phodnota budete vracet p-hodnotu testu

N = 500 # pocet opakovani
rozsahy = c(5,10,50,100,250,500,750,1000) # ruzne rozsahy vyberu
alpha = 0.05 # nominalni hladina vyznamnosti testu
vys = rep(NA,length(rozsahy)) # vektory pro ukladani vysledku
vys2 = rep(NA,length(rozsahy))
vys3 = rep(NA,length(rozsahy))
# sila testu – vyber z t rozdeleni
for (i in (1:length(rozsahy))) { # silu testu studujeme pro ruzne rozsahy vyberu
   mat = matrix(rt(rozsahy[i] * N, df = 10), ncol = N)
   vys[i] = mean(apply(mat, 2, phodnota) <= alpha) }
# vygenerujeme N vyberu, kazdy o rozsahu postupne 10, 100 a 1000, s 10 stupni volnosti
# odhadneme silu testu, uvazujeme nominalni hladinu vyznamnosti alpha
# spocitame, v kolika vyberech je p-hodnota nizsi nez 0.05, a vydelime to poctem vyberu

# sila testu – vyber z uniformniho rozdeleni
for (i in (1:length(rozsahy))) {
   mat = matrix(runif(rozsahy[i] * N, -4,4), ncol = N)
   vys2[i] = mean(apply(mat, 2, phodnota) <= alpha)  }

 # validita testu – vyber z normalniho rozdeleni
for (i in (1:length(rozsahy))) {
   mat = matrix(rnorm(rozsahy[i] * N), ncol = N)
   vys3[i] = mean(apply(mat, 2, phodnota) <= alpha)  }
```

```
# grafy
par(mfrow = c(1,3))
plot(vys~rozsahy, type = "l", main = "Sila testu - vyber z t-rozdeleni")
plot(vys2~rozsahy, type = "l", main = "Sila testu - vyber z uniformniho rozdeleni")
plot(vys3~rozsahy, type = "l", main = "Validita testu - vyber z normalniho rozdeleni", ylim = c(0,1))
```



BOOTSTRAP

Bootstrap je speciální postup, jehož předností je, že **nemusíme znát konkrétní proces generování dat**. Simulací generujeme velký počet tzv. bootstrapových výběrů. Z každého z nich pak spočteme testovací statistiky, jejichž empirická rozdělení porovnáme se skutečnými testovacími statistikami. Tato metoda tedy umožňuje na základě hodnot **jednoho** náhodného výběru odhadovat vlastnosti odhadu parametru (směrodatnou chybu odhadu, vychýlení, intervaly spolehlivosti…).

Představme si například, že chceme získat odhad směrodatné chyby mediánu nebo třeba odhad směrodatné chyby podílu dvou středních hodnot. To už není tak jednoduché jako spočítat odhad směrodatné chyby průměru a navíc to vyžaduje splnění určitých předpokladů. Máme k dispozici jeden výběr z populace. Empirická distribuční funkce je neparametrickým odhadem distribuční funkce v této populaci. Bootstrap považuje tento jeden výběr jako náhražku za celou populaci. Z tohoto výběru o rozhsahu *n* provádíme **opakované výběry s vracením rozsahu *n*** (**resampling**), **a to celkem B-krát**. Tak spočítáme celkem B bootstrapových replikací odhadu $\hat{\theta}_B^*$ (například B odhadů mediánu). Z nich můžeme odhadnout směrodatnou chybu tak, že spočítáme směrodatnou odchylku těchto hodnot.

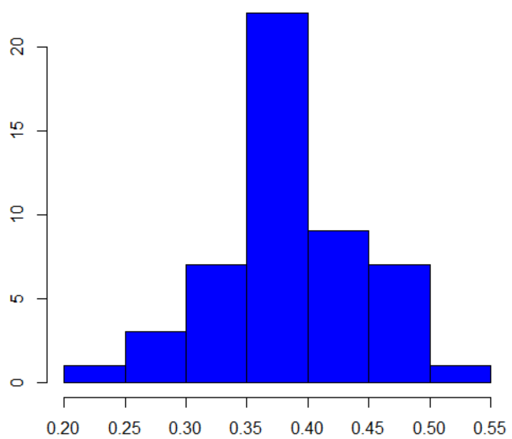
PŘÍKLAD 6: ODHAD SMĚRODATNÉ CHYBY PRŮMĚRU POMOCÍ BOOTSTRAPU

Skript – M.Basta (upraveno)

```
n = 20 # rozsah vyberu
vyb = runif(n) # vybereme 20 hodnot z uniformniho rozdeleni na intervalu (0,1)
B = 50 # udelame 50 bootstrapovych vyberu
mat = matrix(sample(vyb, size = n * B, replace = TRUE), ncol = B)
# v matici mat bude 50 sloupcu a v kazdem sloupci bude 20 hodnot daneho vyberu
brep = apply(mat, 2, mean)  #brep je vektor vyberovych prumeru (prumer hodnot v kazdem sloupci)
(bse = sd(brep)) # bootstrapovy odhad sm.chyby vyberoveho prumeru, tedy sm.odchylka vyberovych prumeru
(sd(vyb)/sqrt(n)) #pro srovnani - klasicky vypocet pomoci bezneho vzorecku
 (se = sqrt(1/12)/sqrt(n)) # skutecna hodnota smerodatne chyby vyberoveho prumeru

hist(brep, col = "blue")  # histogram
est = mean(vyb) # vyberovy prumer puvodniho vyberu jakozto odhad stredni hodnoty
(bISnormD = est - qnorm(0.975) * bse) # tzv. bootstrapovy normalni interval spolehlivosti – dolni hranice
(bISnormH = est + qnorm(0.975) * bse) # tzv. bootstrapovy normalni interval spolehlivosti – horni hranice
# pro intervaly spolehlivosti je vhodne B navysit. Normalni interval nemusi byt uplne nejvhodnejsi volbou.
```

Výsledek mi při jednom konkrétním experimentu vyšel 0,056 pomocí bootstrapu a 0,055 klasickým výpočtem. Skutečná hodnota je 0,065. Histogram výběrového průměru:



PŘÍKLAD 7: BOOTSTRAP V REGRESI

Chtěli bychom nyní odhadnout směrodatnou chybu regresních parametrů. Distribuční funkci náhodné složky můžeme odhadnout pomocí empirické distribuční funkce **reziduí**. Postup spočívá v tom, že z původního vektoru reziduí děláme opakované výběry s vracením. Pro každý bootstrapový výběr spočítáme na základě odhadnutých parametrů, známých hodnot vysvětlující proměnné a těchto resamplovaných reziduí hodnoty vysvětlované proměnné. Pomocí takto spočítaných hodnot vysvětlované proměnné znovu odhadneme parametry regrese. Totéž opakujeme mnohokrát a odhady parametrů si ukládáme. Spočítaná směrodatná odchylka z těchto odhadnutých parametrů je pak odhadem směrodatné chyby parametru.

Skript – M.Basta (upraveno)

```
library(boot)
X = runif(40,0,10) ## vytvorime nejaka data
Y = 2 + 5*X + rnorm(40,0,5)
fit = lm(Y ~ X)  ## odhad parametru metodou nejmensich ctvercu (MNC)
 (sse = coef(summary(fit))[ , 2]) # odhady smerodatnych chyb odhadu, jak je vystupem z regrese

odhadRes = function(rezidua, ind, xprim, odhadyPar) {
  y = odhadyPar[1] + odhadyPar[2] * xprim + rezidua[ind]
  fit = lm(y ~ xprim)
  return(coef(fit))}
  # tato funkce vytvori bootstrapovy vyber
  # na zaklade resamplovani rezidui a nasledne ziska bootstrapovou replikaci odhadu
  # rezidua ... rezidua z primarniho odhadu
  # ind ... indexy pro resamplovani
  # xprim ... hodnoty vysvetlujici promenne pouzite v primarnim odhadu
  # odhadyPar ... odhady regresnich parametru MNC z primarniho odhadu

# bootstrapping, pro napovedu k argumentum funkce viz help(boot)
bo = boot(resid(fit), odhadRes, R = 1000, xprim = X, odhadyPar = coef(fit))
(bseR = apply(bo$t, 2, sd)) # bootstrapove odhady sm. chyb na zaklade resamplovani rezidui
```

MAKROEKONOMICKÁ POLITIKA

HLAVNÍ CÍLE

1. Vysoká a rostoucí úroveň reálného produktu.
2. Nízká nezaměstnanost, vytváření pracovních příležitostí
3. Stabilní nebo mírně se zvyšující cenová hladina s cenami a mzdami stanovenými na volných trzích.
4. Zahraniční ekonomické vztahy, které se vyznačují stabilním měnovým kurzem a vyrovnaným saldem.

MĚŘÍTKA EKONOMICKÉ ÚSPĚŠNOSTI ZEMĚ

1. **Hrubý domácí produkt** = tržní hodnota statků a služeb vytvořená za dané období na určitém území. Rozlišujeme nominální HDP (v tržních cenách) a reálný HDP (ve stálých cenách výchozího období). Potenciální HDP je nejvyšší udržitelný výkon ekonomiky.
2. **Míra nezaměstnanosti**. Nezaměstnaný je člověk v produktivním věku, který si aktivně hledá práci a je schopen do 14 dnů nastoupit do zaměstnání.
3. **Míra inflace** = index růstu či poklesu cenové hladiny. Stanovuje se na základě CPI, tedy indexu spotřebitelských cen, který se počítá z výdajů na pevně stanovený koš statků a služeb.
4. **Saldo obchodní bilance**.

ČR	HDP	nezaměstnanost	inflace	saldo OB
2013	-0,9 % (Q1 2014: +2 %)	7 %	1,4 %	350 mld Kč

NÁSTROJE MAKROEKONOMICKÉ POLITIKY

1. **Fiskální politika**
   a) vládní výdaje - podílí se na tvorbě HNP, stanovuje rozsah veřejného a soukromého sektoru
   b) daně - daně omezují důchody obyvatelstva, napomáhají stanovovat ceny.
2. **Monetární politika**. Zahrnuje regulaci peněz, úvěrů a bankovní soustavy země její centrální bankou. Zrychluje či zpomaluje růst nabídky peněz, snižuje nebo zvyšuje úrokové sazby a povzbuzuje nebo omezuje investice a působí na cenovou hladinu (inflaci).
3. **Zahraniční hospodářská politika**.

a) ovlivňování obchodu prostřednictvím opatření obchodní politiky (cla, kvóty).
b) Regulace měnového trhu (např. nedávné oslabení koruny intervencí ČNB)

4. **Důchodová politika**. Mzdová a cenová politika, souhrn opatření vlády, která usilují o zmírnění inflace pomocí přímých kroků, ať již slovním přesvědčováním, nebo zákonnými regulačními opatřeními, zahrnujícími mzdy a ceny. V posledních letech se již opouští.

HLAVNÍ PROUDY

1) Ekonomie hlavního proudu - neokeynesiánská ekonomie
Za hlavní zlo považují **nezaměstnanost**. Kladou důraz na **fiskální politiku.** Stala se převažujícím směrem po druhé světové válce. Po krizi v 70. letech bylo neokeynesiánství nahrazeno neokonzervativní teorií. V současné době neokeynesiánská ekonomie opět posiluje své postavení.

2) Monetarismus (neoklasická makroekonomie)
Za hlavní zlo považuje inflaci. Kladou důraz na **monetární politiku.** Hlavním nástrojem jsou operace na volném trhu. Vychází z představy, že trhy disponují dostatečnými samoregulačními silami, které jsou schopny navracet tržní ekonomiku bez výraznějších negativních dopadů do stavu rovnováhy. Státní zásahy mají povahu destabilizujících šoků. Za zakladatele se považuje Milton Friedman.

ZDROJE

Hušek, R.: Ekonometrická analýza. Nakladatelství Oeconomica, Praha 2007.

Český statistický úřad. http://www.czso.cz

Zpracované otázky ke státnicím dostupné z
https://drive.google.com/folderview?id=0B5agwzpg7Fj_MWV1dlJBMlM0VEk&usp#grid

Mgr. Milan Bašta, Ph.D.: Cvičení z 4ST417 Výpočetní statistika v R.