STATISTICKÉ PROGRAMOVACÍ JAZYKY

Jiří Hváček, Hana Řezanková, VŠE Praha

Naším cílem je popsat současný přístup k programování statistické analýzy dat z hlediska "programátora" i "uživatele", pokusit se vystihnout to podstatné, co se v této oblasti děje. Čtenář by měl chápat tento příspěvek jako určitý pokus o vytvoření něčeho na způsob "teorie statistických programovacích jazyků", i když spíše ve verbálním smyslu. Některé náznaky podobného přístupu nalezne čtenář i v citované literatuře.

1. Statistické výpočetní prostředí

Pod "statistickým výpočetním prostředím" chápeme tu oblast softwarových produktů, které jsou systematicky používány pro statistickou práci nebo jsou přímo vyvinuty pro potřeby statistiky.

Toto prostředí se dnes dělí (viz např. Thisted, 1986) do tří oblastí:

A) Programové vybavení teoretického statistika, kam lze zařadit zejména:

a) Wordprocessory a grafické systémy, které slouží ke psaní matematických a statistických textů. Patří sem zejména tzv. "vědecké" wordprocessory, jako jsou například TEX, T^3 či i u nás populární ChiWriter, a ty lepší z nich lze použít i pro matematickou sazbu. Velmi výhodné jsou integrované wordprocessory s grafikou a numerikou, jako je například MathCAD. Tato oblast je velmi perspektivní, výrazně zlevňuje a zejména "zpřesňuje" tisk publikací. Vzniká tzv. desktop publishing, publikování "na stole".

b) Pakety a jazyky pro symbolickou matematiku dnes výrazně snižují pracnost některých druhů odvozování (např. derivování, integrování atd.). Jedná se prakticky o první užitečné aplikace umělé inteligence pro vlastní vědeckou práci, z poměrně široké nabídky jmenujme alespoň MuMath či Macsyma.

c) Pakety a jazyky pro numerické výpočty, kam patří dnes zejména nejrůznější systémy podprogramů, ale také novější numerické či maticové jazyky. Kromě souborů procedur jako je třeba IMSL či NAG můžeme jmenovat například systém Eureca nebo částečně i statisticky zaměřené maticové jazyky Gauss či Minitab.

d) Expertní systémy jsou zatím na počátku vývoje. Pro teoretického statistika jsou zajímavé spíše jako oblast aplikace pravděpodobnostní logiky. Patří sem i některé naše produkty, jako je třeba EQUANT a jeho deriváty, v širších souvislostech i GUHA.

B) Programové vybavení pro simulace a Monte-Carlo. Tato oblast se do určité míry osamostatňuje a zejména simulace dnes nabývá poněkud jiného, trochu i filozofického významu. Byly vyvinuty specializované jazyky (např. SIMULA, SIMSCRIPT, GPSS atd.) a obvykle je zapotřebí i vysoce výkonný (nejlépe super-) počítač.

C) Programové vybavení pro analýzu dat, s nímž většina statistiků běžně pracuje (a často i teoretičtí). Z hlediska softwaru sem patří zejména

a) Databankové systémy, které jsou primárně určeny pro archivaci dat a dotazování, ale jejich silnou stránkou je i pořizování dat a lze v nich realizovat i jednodušší statistické výpočty. Ze známějších databankových systému lze jmenovat například dBASE III+ pro počítače PC, nebo SQL pro velké počítače IBM. S rostoucí výkonností počítačů PC postupně dochází ke sbližování databankových systémů velkých a malých počítačů a lze dokonce odhadovat, že standardem se stane patrně SQL (např. avizovaná dBASE IV již má interface k SQL).

b) Tabulkové systémy (spreadsheety) jsou jednak určeny pro uchování dat, která mají charakter dvourozměrných tabulek, jednak pro výpočty v takovýchto tabulkách. Vzhledem k tomu, že zejména v ekonomii (např. finance, účetnictví, kalkulace) mají výpočty vesměs charakter manipulace se sloupci (tak je tomu ostatně do značné míry i ve statistice), jsou velmi rozšířené a řada statistiků realizuje své výpočty přímo v nich. Ze známějších lze jmenovat např. Lotus 1-2-3.

K databankovým a tabulkovým systémům existují i doplňkové statistické produkty, které umožňují realizovat i poměrně složité výpočty (tzv. add-on, např. 123 Forecast!).

c) Statistické programové pakety je podle našeho názoru již nevyhovující, ale přesto používaný název pro softwarové produkty vyvinuté speciálně pro statistickou analýzu dat. Těmi se budeme v dalším textu zabývat.

2. Integrované programové vybavení pro statistickou analýzu dat

Dnes je již zcela zřejmé, že statistickou analýzu reálných dat nelze provádět pomocí jednotlivých, vzájemně neprovázaných počítačových programů. Analýza dat je příliš komplexní, než aby na ni stačil jeden, byť rozsáhlý prostředek.

V zásadě lze každou analýzu dat rozdělit do tří kroků: příprava a verifikace dat, výpočet a prezentace výsledků, rozbor výsledků a případný návrat zpět. Z toho vyplývá, že pro analýzu dat budeme obecně potřebovat flexibilní interaktivní systém, zahrnující blok práce s daty a více bloků realizujících statistické výpočty a jejich prezentaci (včetně grafiky).

Za těchto podmínek se vyvinuly různé integrované prostředky pro statistickou analýzu dat, které můžeme dále dělit na:

a) Soubory statistických podprogramů a procedur, které lze použít v rámci obecných programovacích jazyků. Předpokladem jejich použití je znalost příslušného jazyka a výhodou je možnost využití celého "programovacího prostředí" tohoto jazyka.

b) Soubory statistických programů, které spojuje pouze jednotná metodika (například způsob práce s daty a sjednocení základních příkazů). Určitou nevýhodou je neustálé "opakování" stále stejných struktur v každém programu. S výhodou se však používají tam, kde se stále opakují standardní analýzy (jako třeba v biologii, sociologii atd.).

c) Systémy statistických programů, které jsou založeny na specializovaných modulech pro jednotlivé činnosti. Poznáme je například podle toho, že mají specializovaný modul pro přípravu dat. Tyto prostředky poskytují větší flexibilitu, lze je poměrně rychle rozšiřovat a upravovat a tvoří dnes páteř statistické analýzy dat i pro náročnější analýzy.

d) Statistické makrojazyky, které již mají jednoduchý interpret a dávají dostatečnou flexibilitu i pro velmi náročné analýzy, jako je explorační analýza dat (EDA). Určitou "nevýhodou" těchto prostředků je, že se musíme příslušný makrojazyk naučit, bývá však velmi přirozený a jednoduchý.

e) Statistické programovací jazyky, které vytvářejí programovací prostředí jak pro analýzu dat, tak i pro programování nových metod. I když je to zřejmý trend vývoje, bude tato oblast, podobně jako soubory procedur, vyhrazena specialistům ve výpočetní statistice. Na druhé straně všechny ostatní prostředky zřejmě budou vytvářet právě tito odborníci, takže vývoj v oblasti statistických programovacích jazyků bude klíčový pro všechny ostatní. Z uvedeného důvodu alespoň základní znalosti z této oblasti jsou nezbytné pro každého, kdo bude využívat více prostředků.

Uvedené členění je účelové, pouze z hlediska stupně integrace. Další může být například podle použitého programovacího jazyka (budeme o něm hovořit v odstavci věnovaném procedurám), podle typu počítače (dnes se rozlišují prostředky pro PC a pro velké počítače, ale jsou tendence ke stírání rozdílů) a zejména z věcného hlediska, kde lze hovořit například o

a) obecných statistických programových paketech (patří sem např. BMDP, SPSS, SAS, SYSTAT),

b) maticových jazycích (např. MINITAB, Gauss), v jejichž rámci lze realizovat statistické metody,

c) specializovaných statistických programových paketech.

Tyto specializované prostředky lze dále členit

- podle metody (např. GLIM je pro zobecněný lineární model, dale např. expertní systémy jako REX, STUDENT, tabulkové jazyky jako TPL a podobně),

- věcné, podle oboru, na který je paket specializován, jako
  - systémy pro časové řady a ekonometrii (RATS, SCA),
  - systémy pro analýzu experimentů (XSTAT) a podobně.

Konkrétní prostředek nás v dalším textu bude zajímat již pouze z hlediska stupně integrace.

2.1. Soubory procedur

Prvořadou otázkou při využití souboru procedur je alespoň v současné době otázka programovacího jazyka. Jako jazyk statistických programových paketů převládá jednoznačně FORTRAN zejména proto, že zatím se věnovala pozornost především algoritmické stránce a většina procedur byla publikována a je dostupná právě v něm. Ze známějších souborů statistických procedur je možno jmenovat například IMSL, ACM či NAG.

Prakticky zanikl Algol, poměrně málo se využívá BASIC, a to i přesto, že je všeobecně dostupný - důvodem je zejména neexistence normy a tím daná nepřesnost. Ze stejných důvodů se zatím neprosadil ani Pascal s výjimkou TurboPascalu, který poskytuje vhodné programovací prostředí a je dostatečně rozšířen. Proto se zmíníme o souboru MLIB.

V poslední době se i pro oblast numerických výpočtů prosazuje jazyk C, zejména proto, že je velmi rychlý a snadno přenositelný na různé druhy počítačů.

I když soubory procedur budou zřejmě ještě dlouho k dispozici pouze v klasických procedurálních programovacích jazycích, vlastní programování paketů bude zřejmě prováděno v jazycích umělé inteligence, jako jsou například LISP či Prolog. Podmínkou je ovšem vyřešení implementace přinejmenším přeložených procedur z jiných jazyků.

2.1.1. MLIB

MLIB je poměrně nový a moderní soubor procedur a programů v jazyce TurboPascal 3.0 (připravuje se verze pro TP 4.0), který má dvě podstatné výhody
- procedury jsou speciálně koncipovány pro potřeby statistických výpočtů a konstrukci statistických programových paketů,
- všechny procedury jsou k dispozici ve zdrojovém tvaru, takže je lze případně upravovat pro vlastní potřeby.

Autorem tohoto souboru je Michael Conlon (viz např. American Statistician, 41 (1987), č.4, str.320) a jedná se o tzv. public domain produkt, takže jej lze volně kopírovat (máme jej např. na KST VŠE). Soubor dnes obsahuje řádově 300 procedur a jeho integrující komponentou jsou společné datové typy, zejména dynamická pole pro matice.

Procedury obsahují až na analýzu časových řad prakticky vše, co je pro vytváření paketů a programů třeba - vstup dat včetně menšího spreadsheetu, maticové operace, optimalizaci, generování náhodných čísel, statistické funkce a testy, kontingenční tabulky, grafiku atd.

Součástí souboru je kromě podrobného manuálu (200 stran, je na disketě) programy

MAKE, který "připraví" do zdrojového programu všechny návazné procedury, takže se uživatel může starat pouze o své bezprostřední výpočty,

FTOP, částečný překladač z FORTRANu do TurboPascalu, který provádí inteligentně mechanické úpravy textů FORTRANských programů do tvaru TurboPascalu (bez vstupu a výstupu) - tím se částečně řeší problém většího bohatství FORTRANských procedur a určité koncepční zastaralosti dřívějších verzí FORTRANu.

MLIB je příkladem toho, že i ve "starých" oblastech se mohou objevit nové přístupy.

2.2. Soubory statistických programů

Tato nejstarší oblast specializovaného statistického softwaru má již dnes prakticky jediného významného představitele, paket BMDP.

Pro paket BMDP je charakteristické, že se skládá z mnoha (asi 40) poměrně rozsáhlých programů, které spojuje pouze společný ovládací jazyk. Je možno používat je zcela nezávisle a jedinou "úsporou" je snadnost přechodu na jiný program. Tato forma má své určité výhody - používá se tam, kde je nutno provádět opakovaně standardní komplikované analýzy. Pro současný přístup ke statistické analýze je již tato forma příliš omezující a i z hlediska programového je nesmyslné neustálé opakování stále stejného submodulu, např. pro vstup dat, v každém programu.

V určité podobě se však soubory statistických programů objevují na vyšší úrovni znovu, a to zejména jako forma organizace softwarových depozit.

Softwarový depozit je zpravidla organizován jako databáze s otevřeným přístupem, která má řídící program a databázi programů.

Řídící program organizuje hlavní činnosti, což jsou informace o vlastní činnosti a o obsahu depozita, úprava počítačových médií (např. formátování), přesuny souborů na vlastní médium a případně dokonce přepnutí na výpočet ve zvoleném programu.

Databáze obsahuje pro každý program (nebo systém) podrobnější informaci o programu (citace, popis činnosti, stále častěji manuál, novinky), zdrojový tvar programu či soubory systému (např. data) a přeložený program či soubory.

Vzhledem k tomu, že softwarová depozita jsou nesporně jedním z budoucích trendů i u nás, je třeba věnovat pozornost formulování sjednocujících prvků i pro soubory programů. V USA již existují i národní softwarová depozita (Argonne National Laboratory a Bell Laboratiries).

2.3. Systémy statistických programů

Systémy statistických programů jsou v současné době nejpopulárnějším prostředkem pro analýzu statistických dat, protože jsou "šité" na osobní počítače, a tudíž masově dostupné. Je pro ně typické, že obsahují minimálně dva moduly
- modul pro práci s daty, který obsahuje příkazy pro vstup, kontrolu a transformace dat,
- modul (častěji ovšem více) pro statistické výpočty.

Jako typické představitele můžeme jmenovat např. i u nás rozšířený Statgraphics nebo dnes jeden z nejlepších paketů - SYSTAT. Tyto prostředky zůstávají na úrovni ovládání pomocí menu či jednoduchých příkazů, takže jsou velmi "příjemné" i pro laického uživatele. Jejich výhodou je i "oddělení" funkcí do poměrně málo rozsáhlých modulů, takže mohou být poměrně často upravovány a modernizovány.

2.3.1. SYSTAT

Paket statistických programů SYSTAT je dnes pokládán za jeden z nejlepších. Kromě moderní koncepce je na něm zajímavá např. i originální forma reklamy - jeho "plnokrevná" podmnožina MYSTAT je public domain, a je tedy volně k dispozici.

SYSTAT má modulární architekturu, což znamená, že jednotlivé činnosti jsou sdruženy do relativně "malých" nezávislých modulů, které nemusí být současně v paměti. Komunikace mezi moduly se děje pomocí SYSTATovských souborů. Vzhledem k tomu, že jejich detailní popis je v manuálu (dokonce s texty procedur pro vstup ve FORTRANu), je poměrně snadné psát vlastní moduly - a firma takové moduly dokonce propaguje.

Za charakteristické rysy SYSTATu lze pokládat zejména výborně vyřešený vstup dat a velmi obecně a moderně koncipované statistické moduly (vyznačují se i vysokou rychlostí a přesností).

a) Vstup dat

V rámci SYSTATu lze použít pro vstup dat kromě vstupu z ASCII souborů zejména specializovaný tabulkový editor, který obsahuje

- tabulku, do které doplňujeme čísla a texty a v níž se můžeme pohybovat pomocí kurzorů a dalších kláves,
- příkazový řádek, v němž lze provádět operace nad proměnnými, vytvářet nové proměnné, vyhledávat určité hodnoty atd.

Pro práci s daty jsou dále k dispozici

- speciální datové příkazy, které umožňují např. standardizovat proměnné, nahradit je pořadími, slučovat soubory apod.,
- SYSTATovský BASIC, což je jazyk podobný BASICu, pomocí kterého lze provádět např. generování dat, jejich logickou kontrolu atd.

b) Statistické procedury

SYSTAT je výhodný i pro profesionální použití a výuku statistiků, protože pracuje s velmi obecnými modely. Zdařilé jsou zejména moduly MGLH a NONLIN, ale i ostatní jsou na vysoké úrovni abstrakce.

Práce s každým modulem začíná jeho vyvoláním a vyvoláním vstupního souboru, tedy pro modul MGLH například

```
>MGLH
>USE Jméno_Vstupního_Souboru
```

Práce s modulem je velice jednoduchá. Například modul MGLH (vícerozměrná zobecněná lineární hypotéza) pracuje s vícerozměrným lineárním modelem typu

$$Y = X\beta + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2 I),$$

kde vektor parametrů β se odhaduje (váženou) metodou nejmenších čtverců, a lze v něm testovat hypotézy tvaru

$$A\beta C' = D.$$

V modulu MGLH jsou řešeny dvě základní úlohy:

- definice modelu a odhad vektoru parametrů, např.

```
>MODEL Proměnná = CONSTANT + Proměnná1 + ...
>ESTIMATE
```

- testování hypotéz o vektoru parametrů, např. pro uvedený jednorozměrný regresní model test o nulové hodnotě skupiny parametrů by byl

```
>HYPOTESIS
>EFECT = Proměnná1 & Proměnná2 &...
>TEST
```

2.4. Statistické makrojazyky

Za statistický makrojazyk lze označit některé novější statistické programové systémy, pro které je charakteristické, že v nich lze vytvářet programy a tzv. makra.

Pod možností vytvářet programy chápeme možnost systému vlastními prostředky ukládat posloupnost příkazů, v níž jsou možné cykly, skoky atd. Jako makro rozumíme možnost systému vytvářet podprogramy složené z vlastních příkazů, které je možno volat jménem a používat jako součást systému.

Za statistický makrojazyk lze pokládat například SAS/PC, SCA, ale do značné míry i MINITAB či Gauss. Prakticky všechny současné statistické programové systémy patří do této kategorie (i v SYSTATu je takto ovládán modul DATA), takže název statistické programové pakety je již evidentně historický.

2.4.1. SAS/PC

SAS/PC je dnes zcela jiný produkt než u nás známá starší verze pro velké počítače. Firma změnila svou politiku a pod vlivem komerčního úspěchu jiných paketů a rostoucí výkonnosti PC počítačů připravila novou, uživatelsky "přátelskou" verzi, která je v mnoha aspektech na čele vývoje statistického programového vybavení.

SAS/PC je poměrně rozsáhlý produkt (řádově 6,5 MB paměti) a je součástí širší třídy produktů firmy SAS, která pro verzi PC obsahuje kromě dvou základních modulů Base SAS a SAS/STAT zejména SAS/GRAPH pro grafiku a maticový jazyk SAS/IML. Výhodou je, že verze pro PC i pro velké počítače se liší pouze rozsahem zpracovávaných úloh, takže uživatel je "připraven" i na řešení větších úloh beze změny ovládání.

Základní ovládání je rozděleno do čtyřech oken (windows), v nichž si lze "prohlížet" historii příslušné činnosti a v příkazovém řádku okna vykonávat potřebné činnosti:

OUTPUT, okno výstupů,
LOG, zprávy systému, tzv. deník,
PROGRAM EDITOR, kde zapisujeme příkazy makrojazyka a
HELP, pomoc uživateli.

Pro přechody mezi jednotlivými okny lze použít kurzoru nebo predefinovaných kláves a v každém okně lze provádět např. výpis okna na tiskárně, uložení okna do souboru atd.

Vlastní ovládání SASu spočívá kromě "plnění" příkazů v oknech zejména v zápisu makroprogramu do okna programového editoru (a jeho spuštění). Příkazy makrojazyka mohou obsahovat kromě statistických algoritmů i podmíněné příkazy, cykly, skoky a podprogramy. Práce se SAS/PC je velmi příjemná a skutečnost, že se jedná o ekonomicky velmi silnou firmu a že společně s dalšími produkty prakticky řeší otázku zpracování dat na podnikové úrovni, mu zaručuje slibnou budoucnost.

2.5. Statistické programovací jazyky

Jako statistické programovací jazyky můžeme označit i makrojazyky, kterými jsme se zabývali v předchozím odstavci. Z hlediska computer science se však spíše jedná o "kolekci funkcí vyšší úrovně", uvedené jazyky ještě nemají úplnou formální syntax podobnou známým programovacím jazykům.

V současné době je jediným známějším statistickým programovacím jazykem jazyk S (S language). I když je v našich podmínkách zatím nedostupný, je natolik rozšířen na amerických univerzitách a má tak velký vliv na konstrukci specializovaného softwaru, že jeho znalost je do určité míry podmínkou pro pochopení vývoje v této oblasti.

2.5.1. Jazyk S

Jazyk S sami autoři nazývají "interaktivní prostředí pro analýzu dat a grafiku". I když má velmi silné statistické algoritmy, je určen převážně k programování nových. Vznikl v Bellových laboratořích v letech 1983-84. Vyvinuli jej Chambers s Beckerem, je psán v jazyce C a pracuje v prostředí operačního systému UNIX. Kromě i u nás známé knihy vyšla i další, pojednávající o implementaci procedur psaných v jazyce C a FORTRANu a v současné době je ověřována jeho další verze (1987).

Jazyk S pracuje interaktivně (příkazy se okamžitě provádějí), ale lze přirozeně pracovat i v dávkovém režimu (používá se zejména výpočtu "v pozadí"). Syntax jazyka S je velmi podobná obecným programovacím jazykům, nalezneme zde konstanty, proměnné (které zde reprezentují celé struktury), operátory; můžeme vytvářet výrazy, cykly, podprogramy atd. Součástí jazyka je i velmi široká škála maticových, statistických a grafických funkcí. Možnost vytvářet vlastní procedury (tzv. makra) vedla ke vzniku řady specializovaných knihoven programů a maker pro nejrůznější aplikační oblasti.

3. Prostředky ovládání programového vybavení

Prostředky ovládání softwaru, tedy i statistických programových paketů, lze rozdělit do čtyř skupin, které odpovídají v běžné lidské komunikaci pojmům slovo, věta, odstavec a rozhovor (vzdáleně). Můžeme rozlišit:

A) Výběr z nabídky možností, které jsou obecně vhodné pro ovládání hierarchických struktur. Patří sem zejména menu, klíče, ikony, okna a klíčová slova. Jejich výhodou obecně je jednoduchost a dále skutečnost, že uživatel dostává s nabídkou zpravidla i základní textovou či grafickou informaci o tom, co může v dané situaci dělat.

B) Strukturované příkazy, jejichž jednou formou je tzv. panel. Jsou dnes nejužívanějším prostředkem ovládání statistických algoritmů, protože odpovídají struktuře většiny statistických procedur a vyhovují interaktivnímu zpracování.

C) Jazykové prostředky, kam zahrnujeme makrojazyky a programovací jazyky. Zatímco makrojazyky mají pouze minimální syntax, v programovacích jazycích můžeme deklarovat nové objekty a procedury.

D) Dialog, který na rozdíl od předchozích prostředků předpokládá od počítače nejenom prosté porozumění, ale také určitou aktivitu - počítač se může dotázat na chybějící informaci, podat vysvětlení atd.

Konkrétní programový systém již dnes používá zpravidla více prostředků, takže členit software podle způsobu ovládání je do značné míry beznadějné.

3.1. Výběr z nabídky možností

a) Menu (česky někdy jídelníček) je velmi významný nástroj zejména pro laického uživatele. Je efektivním nástrojem pro ovládání hierarchických struktur a nezanedbatelná je i jeho popisná funkce - při procházení menu dostává uživatel i základní informace. I když se menu pokládá za nejprimitivnější formu ovládání softwaru, využívají jej i nejmodernější jazyky (v S existuje dokonce i funkce na vytváření menu). Vzhledem k tomu, že s častějším užíváním rostou znalosti uživatele a také statistická analýza nemá vždy ryze hierarchický charakter, nejsou pakety, které se ovládají pouze prostřednictvím menu, pro častého uživatele příliš atraktivní.

b) Klíče (keys) jsou velmi efektivní "zkratkou" pro často užívané příkazy, je ovšem nutno používat je s mírou a pokud možno v konzistenci se standardním užíváním klíčů v ostatním softwarovém vybavení. Kromě běžných "PC" klíčů se také požívá zkratek příkazů či kombinací kláves. V každém případě by se měly dodržovat tyto zásady:
   - definice klíčů by měly být v každém okamžiku přítomny na obrazovce,
   - měly by se používat v běžném významu (jako např. Esc pro "záchranu" a přechody na vyšší hierarchickou úroveň).

c) Ikony jsou vymezené plochy na obrazovce, obsahující pro rozlišení zpravidla výstižný obrázek (odtud název). Lze do nich "najet" kurzorem či myší a tím vyvolat příslušnou činnost. Jsou populární zejména u laických uživatelů a předpokladem použití je zejména vysoká rozlišovací schopnost obrazovky. Ve statistice se využívají např. pro výběr grafů (paket 3D).

d) Okna (windows) jsou ve skutečnosti spíše aspekty výpočtu, mají tedy poněkud hlubší smysl než pouhý výběr z nabídky možností v "postupném" výpočtu. Zpravidla se v nich realizuje určitá činnost (v paketu SAS/PC jsou např. 4 okna - pro vstup dat a řízení výpočtu, deník, výsledky a help), do které můžeme přejít, pokračovat v ní a vystoupit z ní, aniž bychom ztratili obsah ostatních oken. Jsou to tedy skutečná okna, různé pohledy na mnohotvárnou činnost programu. Ovládání oken a ikon je v mnohém velmi podobné, kromě "najetí" je to zejména zvětšení (zoom) a zmenšení (unzoom), v okně můžeme navíc listovat (scroll).

e) Klíčová slova jsou v podstatě pouze mnemotechnickou zkratkou za položky menu. Jejich výhodou však je, že nemusíme stále pozorovat obrazovku a v případě, že je menu příliš strukturované, můžeme podstatně zkrátit i proces procházení více obrazovkami.

3.2. Strukturované příkazy

Strukturované příkazy mají za cíl realizovat určitou činnost počítače. Na rozdíl od výběru z nabídky možností zde má uživatel určitou volnost, platí ovšem za ní tím, že musí vědět, co se od něj očekává. Příkazy lze rozdělit na:

- Specifikační, které pouze vymezují příslušnou činnost (je to něco na způsob "hlavičky" a obvykle je to buď metoda nebo přímo modul paketu). U současných počítačů se v tomto okamžiku zpravidla nahraje do vnitřní paměti příslušný úsek programu.
- Přípravné a modifikační, které upravují určité struktury v počítači (deklarace modelu, proměnných apod.). Vzhledem k tomu, že se zdánlivě "nic neděje", používá se někdy pro tyto příkazy označení "cold".
- Výkonné, tzv. "hot", které "spouštějí" příslušný výpočet či činnost.

U dnešních paketů se zpravidla používá celá tato struktura (SAS/PC napřed nahraje data, což je posloupnost "cold" příkazů uzavřena mezi DATA a výkonným RUN, podobně analýza je mezi PROC a RUN).

O strukturovaném příkazu zde hovoříme proto, že má složitější strukturu (viz analogie věty), jak ukazuje i následující příkaz posloupnosti strukturovaných příkazů paketu SYSTAT, realizující odhad parametrů logistické křivky metodou nejmenších modulů:

```
>NONLIN
>USE Datový_Soubor_Obsahující_Proměnnou_Y
>MODEL Y = c/ (1 + b*a↑CASE)
      zde c, b, a jsou parametry, CASE je pořadí (index) pozorování
>LOSS = ABS(Y - ESTIMATE)
      zde ESTIMATE jsou teoretické hodnoty Y
>ESTIMATE
```

V příkladu je NONLIN specifikační příkaz, MODEL a LOSS přípravné a USE a ESTIMATE výkonné příkazy. Na příkladu též vidíme, že formulace úlohy může zároveň sloužit jako popis - i ten, kdo SYSTAT nezná, se dovtípí i bez vysvětlení, co se zde počítá.

Velmi účinný nástroj pro ovládání strukturovaných příkazů je panel. Jeho výhodou je, že obsahuje celou strukturu příkazu i se základním vysvětlením a také případné default hodnoty, které jsou zároveň i určitým příkladem, co máme případně dosadit. Pro panel je také výhodné, když se můžeme ve výpočtu vrátit zpátky. Tato vlastnost je podle našich zkušeností zcela nezastupitelná, např. při grafickém znázorňování.

4. Hlavní komponenty statistických programovacích jazyků

Jako statistický programovací jazyk chápeme takový specializovaný prostředek ovládání softwaru pro statistickou analýzu dat, který má alespoň některé rysy obecného programovacího jazyka ve smyslu computer science. Za takové rysy pokládáme např.

- syntax, možnost vytvářet objekty (data či výpočty) podle určitých pravidel a ne pouze možnost výběru z určitého seznamu,
- extenzibilitu, možnost vytvářet nové typy objektů,
- konzistenci, jednotnost pojmů a označení s problémovou oblastí,
- portabilitu, nezávislost na typu počítače.

Za základní tendence ve vývoji statistických programovacích jazyků lze pokládat:

- jde o počítačový jazyk konzistentní se statistikou,
- respektuje se existence dalšího softwaru,
- uplatňuje se souborově orientovaný přístup, komunikace mezi jednotlivými moduly a prvky jazyka a okolím se děje prostřednictvím souborů.

Vlastní statistický programovací jazyk lze rozdělit do pěti komponent (příp. oken)

A) vstup dat,
B) řízení výpočtu, statistické algoritmy a prostředky pro realizaci statistických algoritmů,

C) výstup, grafická a textová úprava výstupu,
D) pomoc (HELP), autodeskriptivní komponenta jazyka, jejíž význam je dnes často rozhodující pro použití jazyka,
E) deník (diary, journal), záznam historie výpočtu, který je nezbytný pro seriózní práci a bývá často doplňován výstupem.

4.1. Vstup dat

Základním pojmem ve vstupu dat je tzv. datová matice, což je matice, jejíž sloupce jsou proměnné (ukazatele) a řádky pozorování (případy, objekty).

Programové prostředky pro řízené vstupu dat lze rozdělit do tří základních skupin, které nazveme podle nejčastěji používaných příkazů pro jejich inicializaci. Lepší systémy jich obvykle používají více. Rozlišujeme následující možnosti vstupu dat:
a) Interaktivní vstup jednotlivých hodnot (INPUT) či celého datového souboru (READ) z klávesnice. Stále častěji se užívá specializovaný tabulkový editor (spreadsheet), v jehož rámci lze vytvářet další proměnné a provádět jednoduché výpočty.
b) Vstup dat z vlastních souborů systému (USE).
c) Vstup ze souborů jiných systémů (IMPORT).

V pokročilejších statistických programovacích jazycích se dnes již většinou předpokládá, že soubor byl vytvořen jiným programovým prostředkem. Minimálním požadavkem je umožnění vstupu z ASCII souborů, často se užívá i souborů WordPerfectu, WordStaru, dBASE a Lotusu.

Ve vstupu dat se používají dále
a) Prostředky pro manipulaci s daty, které mají zpravidla charakter specializovaného jazyka. Pomocí jeho příkazů lze vytvářet nové proměnné, jež jsou logickou a aritmetickou kombinací předchozích proměnných, a také kontrolovat data.
b) Prostředky pro generování dat (zejména hodnot nejrůznějších náhodných veličin).

4.2. Řízení výpočtu

Statistické programovací jazyky pracují ve specializovaném prostředí a tomu se musí do jisté míry přizpůsobit i struktura jazyka, který má své určité zvláštnosti. Především je nutno uvést, že statistický programovací jazyk je primárně určen pro přímý výpočet, bude tedy interaktivní. Základní prvky jazyka budou vždy obsahovat příkazy
- deskriptivní, které popisují data a modely,
- výkonné, které realizují statistickou proceduru,
- řídící, které ovládají posloupnost výpočtu.

4.2.1. Deskriptivní příkazy

Deskriptivní příkazy popisují zejména data, dále struktury, které vznikají při výpočtech a v novějších jazycích i používané modely, čímž se statistické programovací jazyky již blíží vlastnímu (neprogramovacímu) jazyku statistiky.

Základním deskriptivním příkazem je deklarace, kterou lze provádět
- Explicitně, přímou deklarací objektu, která se však podstatně liší od deklarace v obecných programovacích jazycích v tom, že nepopisuje rozsah, ale pouze strukturu objektu. Znamená to, že máme k dispozici i specializované operace, které formálně objekt nemění (používá se stejně), ale mění jeho rozsah: např. příkazy typu NEW, které objekt rozšiřují, jiné pro vypouštění komponent, vybírání atd.
- Implicitně, odvozením vlastností objektu jako výsledku operací; tento způsob deklarace je ve statistických programovacích jazycích zcela převažující.

V případě deklarace modelu musí jazyk také "poznat", zda komponenty modelu jsou data nebo parametry.

Jako příklad si uveďme deklaraci modelu v SYSTATu, kde v příkazu

MODEL Y = k/(1+a*b^T) + c

bude v případě, že v datovém souboru jsou obsaženy proměnné Y a T, výsledkem model posunuté logistiky a nově budou deklarovány parametry k,a,b,c, zatímco v případě, že v datovém souboru se nacházejí proměnné Y,a,b,T, tak se jedná o hyperbolu!

Pro programátora je do jisté míry šokující, že statistické programovací jazyky obvykle nerozlišují mezi strukturou a jejím prvkem, podle kontextu se "dosadí" správný význam. Matice a její prvek se mohou například používat pod stejným označením.

4.2.2. Výkonné příkazy

Výkonné příkazy realizují statistickou či jinou operaci nad daty. V terminologii programovacích jazyků se jedná vlastně o procedury, které obsahují jméno příkazu, specifikaci vstupu, specifikaci výstupu, restrikce a opce (options), pod kterými chápeme další modifikující parametry příkazu.

Terminologie není přirozeně jednotná, jako příklad si uvedeme zadání dvourozměrné regrese v jazyce S:

reg (x, y[y>0]) $resid

realizuje regresní analýzu (včetně výstupů) mezi proměnnými x a y pro kladná y a uloží residua pro další analýzu.

Je všeobecná tendence ke stále "abstraktnějším" příkazům, které jsou adekvátní celým skupinám statistických metod. Jako příklad si uveďme dvojici příkazů MODEL a LOSS v modulu NONLIN v SYSTATu, kde různými variantami modelů a ztrátových funkcí dostaneme např. nelineární regresi, částečnou regresi, robustní regresi atd. Podobně je tomu v jiných oblastech, takže například lineární regrese, diskriminační analýza, analýza hlavních komponent atd. se realizují jako obecný lineární model.

Tato tendence má dopad i na statistickou metodologii a výuku statistiky, je třeba hledat vhodné příkazy a jazyky pro popis statistických metod, které by byly srozumitelné jak uživateli, tak i počítači.

4.2.3 Řídící příkazy

V některých situacích je výhodné realizovat výpočet nikoliv jako posloupnost interaktivních kroků, které se okamžitě realizují, ale vytvářet něco na způsob jednoduchých programů. Tyto možnosti, jako je například jednoduché opakování, dnes poskytují i ty nejjednodušší statistické pakety. Velmi užitečná je zejména možnost uložit si posloupnost příkazů do externího souboru a opět jej vyvolat např. s jinými daty.

Z celé řady možností, jak realizovat složitější výpočty, jmenujme alespoň:

a) Podmíněné příkazy, které umožňují vykonávat výpočet pouze při splnění určitých podmínek (např. v závislosti na datech).

b) Cykly, které umožňují realizovat různé druhy opakovaných činností (ve statistických paketech se velmi často vyskytuje konstrukce

Příkaz BY Kategorická_Proměnná ,

která je velmi užitečná např. při analýze kontingenčních tabulek).

c) Podprogramy, jako jednoduchý způsob opakování části programu.

d) Procedury, které se obvykle nazývají makra a které mají podobné vlastnosti jako v obecných programovacích jazycích. Liší se od nich zpravidla tím, že není třeba uvádět všechny formální parametry - v případě neuvedení se dosadí default hodnoty.

4.2.4. Extenzibilita

Po určité době používání systému nutně dospějeme do stádia, kdy se nám začne zdát systém omezený a jeho rozšiřování vlastními prostředky jazyka přestane vyhovovat.

I pro jazyk S, kde je funkce MACRO zdánlivě všemocná, začneme po čase vidět, jak je realizace složitějších algoritmů těžkopádná a zejména pomalá - a že by bylo mnohem výhodnější si napsat tutéž proceduru v jiném programovacím jazyce či převzít zdařilou proceduru odjinud. V neposlední řadě je třeba každý systém neustále rozšiřovat vzhledem ke konkurenci a je tedy vhodné racionalizovat i tuto činnost.

V současné době se používá dvou přístupů - rozšiřování pomocí společných datových souborů a možností implementace procedur z jiných programovacích jazyků.

a) Extenzibilita pomocí společných datových souborů vyžaduje
   - modularitu systému, musí existovat modul pro práci s daty,
   - přesnou definici, dokumentaci a malou variabilitu běžných datových souborů, při dodržení alespoň minimálního společného ovládání modulů.

   Tento přístup k rozšiřování možností systému využívá například SYSTAT, ve kterém je přímo v manuálu uvedena přesná definice datových souborů, texty FORTRANských procedur pro práci s datovými soubory a nabídka popularizace externě vytvořených modulů.

b) Implementace procedur z jiných jazyků přímo do jazyka je zatím omezena pouze na jazyk S a je poměrně složitá (Becker s Chambersem ji věnovali celou monografii). Vyžaduje vytvoření určitého algoritmického jazyka a kompilaci nové funkce do prostředí S.

4.3. Výstup výsledků

Výstup je v zásadě závislý na charakteru příslušné statistické procedury, ale přesto existují dva základní přístupy a celá řada speciálních operací ovládajících výstup.

a) Přímý výstup na konkrétní medium je dnes již zastaralý přístup, používá se prakticky pouze výstup na obrazovku s možností "hardcopy" u jednodušších paketů.

b) Výstup do souboru, který se paralelně objevuje na obrazovce. Tento způsob dnes již zcela převažuje, u lepších paketů je k dispozici i příslušné okno se speciálními operacemi pro listování ve výstupním souboru (scroll), editování a výstup souboru na nejrůznější přídavná zařízení. Součástí výstupu je i identifikace souboru. Používá se například datum a čas, poznámky v průběhu interaktivního výpočtu a stručná informace v pravidelných intervalech (např. v hlavičce nové stránky, tzv. stamps).

4.4. Pomoc uživateli

Help, česky pomoc či vysvětlení, je dnes jednou z nejdůležitějších funkcí pro libovolný software. Vzhledem k tomu, že paměti již přestaly být závažným problémem a texty manuálů se stejně píší na wordprocessorech, tak je jenom přirozené, že se dodávají manuály včetně softwaru na počítačovém mediu. Vzhledem k dynamice vývoje je to jediné racionální řešení. V zásadě lze help rozdělit do čtyř skupin:

A) Informační soubory či programy, což jsou zejména
   - README (česky se skutečně používá CTIME), základní informace o instalaci,
   - TUTORIAL, dnes stále častěji výukový program než stručná učebnice,
   - MANUAL, podrobnější informace programátorského typu,
   - NEWS, novinky, změny oproti předchozí verzi.

B) Vlastní HELP, který může být organizován
   - hierarchicky, kdy bývá přehled možností často kombinován s menu (je to výhodné pro laiky a ty, kteří systém nepoužívají příliš často);
   - interaktivně, a to buď
     - dotazem, což je vlastně příkaz jazyka s příslušnými parametry,
     - přechodem do režimu help a odpovědí z kontextu (používá se nejčastěji kláves - např. Alt-H a Esc pro návrat).

Interaktivní help do značné míry nahrazuje TUTORIAL i MANUAL (často jsou příslušné textové soubory totožné).

C) ERRORS, chybová hlášení, která se objeví současně se stručným vysvětlením chyby a návodem, jak pokračovat dál. Je výhodné ukládat jak chybová hlášení, tak helpy do externích souborů a to nejenom proto, že i v našich podmínkách to usnadňuje překlad softwaru do češtiny - je tak možno například doplňovat potřebné informace na základě vlastních zkušeností.

D) WORKSPACE, stav pracovního prostoru jazyka, který obsahuje např. informace o proměnných, hodnotách proměnných, modelech atd. Kromě dotazování na okamžitý stav workspace je také výhodné, když je možno uložit celý workspace tak, aby byla možnost po přerušení pokračovat od určitého místa výpočtu.

Help do značné míry spoluvytváří rozhraní mezi uživatelem a systémem. Stále více se používá alespoň těch nejjednodušších principů umělé inteligence a při znalosti základních principů práce příslušného softwaru můžeme se současnými systémy pracovat i bez jakéhokoliv studia.

Bohužel jednu z podmínek efektivního využívání helpu, a tím i jakéhokoliv současného softwaru, ani ten nejlepší help v dohledné době neodstraní - a tou je znalost angličtiny. Snažit se překládat helpy a dokumentaci je typická sysifovská práce a v tomto smyslu je snad vhodnější i používání jednotlivých anglických termínů. I při nejlepším překladu ztrácíme totiž jeden z podstatných rysů - interpretaci příkazů (např. překlad helpu slůvkem pomoc stejně zachová mnemotechnický Alt-H).

4.5. Deník

Deník (v angličtině diary, ale též log nebo history) je poměrně málo známou funkcí, která se ve dřívějších statistických programových paketech nevyskytovala. Jeho význam však v poslední době roste, někteří autoři (Chambers) vidí v určitém zobecnění této funkce podstatné zlepšení procesu interaktivní analýzy dat.

Deník vytváří soubor, který obsahuje datum a čas a postupně všechny příkazy, které uživatel zadává systému a lze do něj zapsat i komentáře a přidat i všechny výstupy, takže k dispozici je úplná dokumentace celé analýzy.

Pomocí wordprocessoru lze deník dále upravovat (tak například vznikl velmi pohodlné manuál pro S). Další použití je pro rekapitulaci výpočtu (pouze se vhodně upraví příkazy a celý výpočet se může "jet" znovu, např. s jinými daty).

Tuto funkci má i SYSTAT, příkazem OUTPUT COMMANDS se uloží deník do souboru COMMANDS.DAT (jinak se při výstupu ze systému celý deník zapomene).

5. Závěr

Ke konci si shrňme několik základních tendencí ve vývoji statistického programového vybavení pro analýzu dat. Základní tendence jsou podle našeho názoru dvě:

a) Komercializace, která má za následek zejména profesionální kvalitu produktu, existenci dalších návazných služeb jako je zejména neustálá inovace a přizpůsobení nejnovějším podmínkám, nejrůznější "poradenské" služby atd. Na druhé straně je to i určitá změna od dosavadní praxe "bezcenného softwaru", uživatelé si musí konečně uvědomit, že jednak nemá smyslu konkurovat profesionálům vytvářením vlastního softwaru (a měli by se zaměřit na to, co dosud na trhu chybí), jednak že za software se musí zaplatit (případně nést následky za krádež jako je tomu u jiného zboží). Současné produkty jsou vzhledem k jejich kvalitě ve skutečnosti velmi levné.

b) Existence zobecněného prostředí, v němž se řada funkcí realizuje pomocí jiných programových prostředků (zejména databází, tabulkových procesorů a hlavně wordprocesorů). Ignorování tohoto vývoje a pokusy o vytváření vlastních originálních prostředků nutně vedou k neúspěchu. V tomto směru je nutno se přinejmenším zmínit o

SAA (System Application Architecture) firmy IBM, jejímž cílem je vytvořit jednotné uživatelské prostředí pro všechny třídy počítačů - programy se budou chovat stejně na "písičkách" jako na velkých počítačích. Tento program je již realizován v novém operačním systému OS/2, takže uživatelé si prostě vynutí to prostředí, na které jsou zvyklí.

Pro statistiky jsou důležité dvě další tendence

c) Tendence ke statistickému makrojazyku. Jazyk počítače se stále více přibližuje "jazyku statistiky", způsob zadání statistických úloh je stále "přirozenější" - zadáme např. pouze data, model a ztrátovou funkci. Znamená to přinejmenším dvě věci
   - je třeba znát základní pojmy mnohem většího okruhu úloh, a to na "matematické" úrovni,
   - je třeba vytvářet vhodné statistické makrojazyky pro jednoduchý popis úloh.

d) Tendence ke zobecňování, která vede k tomu, že zadávání úloh je stále "obecnější", že dříve zcela rozličné úlohy se řeší formálně stejným způsobem (jako např. nelineární regrese a robustní regrese v modulu NONLIN nebo lineární regrese, ANOVA a diskriminační analýza v modulu MGLH paketu SYSTAT). Znamená to, že u statistických úloh musíme hledat nejenom věcné, ale i formální shody - což možná povede ke kvalitativně novým statistickým metodám.

Projevuje se ještě další všeobecná tendence

e) Aplikace umělé inteligence. I když dnes ještě nemůžeme být spokojeni s dosaženým stavem vývoje v této oblasti, některé aplikace se rozhodně projevují. Jsou to
   - používání "inteligentních" HELPů na základě kontextu,
   - používání jazyků umělé inteligence pro programování paketu (algoritmická stránka jazyků dnes poněkud ustupuje),
   - nasazení některých jednodušších prostředků umělé inteligence přímo do statistických algoritmů (např. užití symbolické matematiky v nelineární regresi odstraňuje nutnost zadávání derivací a podobně),
   - expertní systémy se postupně stávají součástí softwarových prostředků (tato oblast je teprve ve stádiu vývoje).

Otázkou je přirozeně, jak máme na tyto tendence reagovat. Z hlediska výuky je zřejmě nutno rozšířit škálu vyučovaných metod. Je absurdní, že absolvent vysoké školy nezná většinu statistických metod obsažených v nejlevnějším paketu. Znamená to ovšem zvládnout nejenom jejich verbální popis, ale i určitý "matematický nadhled" - jejich matematické zvládnutí z hlediska formulace (nikoliv výpočtu). Je třeba respektovat také tu skutečnost, že vysoká škola je místo, kde se potenciální uživatel poprvé s paketem setkává - a že bude mít přinejmenším tendenci tento první paket používat velmi dlouho.

Je tedy velmi důležité zvolit vhodný paket pro výuku a pokud možno jej sjednotit s tím, co bude dále používat na svém dalším pracovišti. Je pochopitelné poněkud předčasné zaujímat v našich podmínkách stanovisko k této problematice, ale přesto se k ní vyjádříme: Domníváme se, že tímto "řadovým" paketem by měl být SAS/PC. Pro výuku statistiků - specialistů by byl vhodnější spíše SYSTAT.

Literatura:

Becker, R.A.- Chambers, J.M.: S: An interactive enviroment for data analysis and graphics. Wadsworth, 1954.

Extending the S system. Wadsworth, 1985.

Francis, I.: Statistical software. A comparative review. North Holland, 1951.

Silvestrov, D.S.: Programnoje obespečenije prikladnoj statistiky. Moskva, 1956.

Thisted, R.A.: Computing enviroment for data analysis. Stat.Sci.,1956, č.1, str.259-75.

Řezanková, H.-Žváček, J.: Pakety statistických programů. Statistika, 1957,č.1,str.100-75.

Řezanková, H.: Statistické programové pakety pro osobní počítače.Statistika,1957,č.12, str.557-61.