

CLAN

- ett SAS-program för skattningar av medelfel

Claes Andersson
Lennart Nordberg



R&D Report
Statistics Sweden
Research - Methods - Development
1992:11

Från trycket
Producent
Ansvarig utgivare
Förfrågningar

Augusti 1992
Statistiska centralbyrån, utvecklingsavdelningen
Lars Lyberg
Claes Andersson, 019-17 64 97
Lennart Nordberg, 019-17 60 12

© 1992, Statistiska centralbyrån
ISSN 0283-8680
Garnisonstryckeriet, Stockholm

INLEDNING

TILL

R & D report : research, methods, development / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1988-2004. – Nr. 1988:1-2004:2.

Häri ingår Abstracts : sammanfattningar av metodrapporter från SCB med egen numrering.

Föregångare:

Metodinformation : preliminär rapport från Statistiska centralbyrån. – Stockholm : Statistiska centralbyrån. – 1984-1986. – Nr 1984:1-1986:8.

U/ADB / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1986-1987. – Nr E24-E26

R & D report : research, methods, development, U/STM / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1987. – Nr 29-41.

Efterföljare:

Research and development : methodology reports from Statistics Sweden. – Stockholm : Statistiska centralbyrån. – 2006-. – Nr 2006:1-.

R & D Report 1992:11. CLAN – ett SAS-program för skattningar av medelfel / Claes Andersson, Lennart Nordberg.
Digitaliserad av Statistiska centralbyrån (SCB) 2016.

CLAN

– ett SAS-program för skattningar av medelfel

**Claes Andersson
Lennart Nordberg**

Innehåll

	sid
0. Abstract	2
1. Inledning	3
2. Vilka situationer klarar CLAN?	4
2.1. Några exempel	4
2.2. Designer och parametrar	5
2.3. Jämförelse mellan CLAN och SMED	7
2.4. Jämförelse mellan CLAN och SUDAAN	7
3. Användargränssnitt	8
4. Hur gör man?	13
4.1. Allmänt	13
4.2. Exempel	14
4.3. Tabeller	19
5. Kapacitet prestanda och felmeddelanden	20
5.1. Test 1	20
5.2. Test 2	21
5.3. Test 3	21
5.4. Felutskrifter	22
6. Formelapparaten	23
6.1. Allmänt	23
6.2. En användbar transformation	24
6.3. Stegvis användning av Woodruff-transformationen för sammansatta funktioner	25
6.4. Variansformler under strategi 1–4	27
7. Referenser	29

0. Abstract

CLAN är ett program för beräkning av punkt- och medelfel-skattningar.

CLAN är programmerat i SAS och kan användas både i PC och stordator.

Låt $X_1, \dots, X_q, \dots, X_Q$ vara ett godtyckligt antal totaler som får avse helt godtyckliga redovisningsgrupper $d_1, \dots, d_q, \dots, d_Q$ vilka får vara varandra överlappande på ett godtyckligt sätt, dvs ett objekt får tillhöra godtyckligt många av $d_1, \dots, d_q, \dots, d_Q$.

Låt $\hat{X}_1, \dots, \hat{X}_q, \dots, \hat{X}_Q$ vara vanliga (Horvitz-Thompson)-skattningar av totalerna $X_1, \dots, X_q, \dots, X_Q$.

CLAN kan användas för att skatta medelfelet för $\hat{Y} = f(\hat{X}_1, \dots, \hat{X}_q, \dots, \hat{X}_Q)$ där f är en godtycklig rationell funktion.

CLAN kan i samma körning behandla ett godtyckligt antal rationella funktioner där man för varje funktion i princip kan använda olika totaler och/eller redovisningsgrupper. Styrkan i programmet ligger i de stora möjligheter användaren har att kombinera val av funktioner och komplexa specifikationer av redovisningsgrupper.

Förändringsskattningar vid paneldata och skattningar som utnyttjar hjälpinformation är några tillämpningar där CLAN erbjuder helt nya beräkningsmöjligheter jämfört med tidigare existerande programvara vid SCB. Även mer elementära fall kan givetvis hanteras med CLAN som bland annat klarar alla typer av skattningar som t ex SMED83 för närvarande täcker.

1. Inledning

Variansberäkningar är en av grundstenarna i kvalitetsarbetet vid SCB. Det finns ett GD-beslut (1986-12-03) som innebär att variansberäkningar skall göras för alla urvalsundersökningar så snart det är tekniskt möjligt. Vi har sedan länge tillgång till ett antal variansprogram. SMED 83, ett SCB-tillverkat program för stordator, och SUDAAN, ett USA-tillverkat program för PC, är de bäst dokumenterade och spridda. Dessutom finns ett antal program, TANJA, VANJA, VP3, EXTRACT, m.fl. som tagits fram internt, för I-avdelningens behov.

Trots denna arsenal av programvara kvarstår ändå vissa allvarliga luckor. De ovannämnda programmen ger exempelvis mycket begränsade möjligheter att beräkna varianser för förändringsskattningar när man har paneldata. Men det finns också andra luckor. Vi har exempelvis hittills haft mycket små möjligheter att göra variansberäkningar för en del vanliga estimatorer som utnyttjar hjälpinformation. Ett sådant exempel beskrivs nedan.

CLAN som skrivits i SAS, har tagits fram för att kunna hantera dessa och många andra fall. Det innebär inte att CLAN kan hantera varje tänkbart variansberäkningsproblem som kan uppstå. Det är inte rimligt att driva arbetet med ett standardprogram så långt. Det kommer alltid att finnas vissa fall som kräver skraddarsydda lösningar.

Tillkomsten av CLAN innebär inte heller att de andra programmen, SMED, SUDAAN etc. blir överflödiga. CLAN kan visserligen hantera (i stort sett) varje situation som något av de övriga programmen klarar men det finns vissa fall av enklare designer och skattningar där CLAN inte är det CPU-mässigt effektivaste programmet. Vi återkommer till de här frågorna senare. Vi menar dock att CLAN bör bli ett värdefullt tillskott till SCB:s verktygslåda för variansberäkningar.

I denna rapport beskrivs CLAN, dess möjligheter och begränsningar. I avsnitt 2 redogör vi för de urvalsdessigner, bortfallsmodeller och skattningar som CLAN kan användas till samt antaganden om övertäckning. Dessutom jämförs CLAN med SMED83 och SUDAAN. I avsnitt 3 beskrivs principerna bakom programmets arbetssätt och användargränssnittet. I avsnitt 4 visas med ett antal exempel hur man löser ett antal vanliga estimationsproblem med CLAN. I avsnitt 5 visas några exempel på CLANs prestanda i stordator och PC-miljö med avseende på CPU-tid. I det sammanhanget görs jämförelser med SMED83 och SUDAAN med avseende på tidsåtgång. Avsnitt 5 innehåller också en beskrivning av hur CLAN hanterar syntaxfel som användaren åstadkommit. Det sista avsnittet 6 kan betraktas som ett tekniskt avsnitt som innehåller formelapparaten som CLAN arbetar efter.

Normalanvändaren torde behöva läsa avsnitt 1-5 medan avsnitt 6 kan sägas vända sig till den intresserade metodstatistikern.

2. Vilka situationer klarar CLAN?

Vi börjar med att ge tre exempel på variansberäkningar som kan göras med CLAN men däremot inte med något av de övriga program som nämndes i avsnitt 1.

2.1. Några exempel

Exempel 1. Låt T_{ij} och N_{ij} vara totalinkomst resp. antal personer i cell (i, j) i följande tabell, (index i svarar mot socialgrupp och j mot åldersgrupp). Vi antar att N_{ij} är okända från början och måste skattas från urvalet.

Socialgrupp	Ålder			
	-29	30-44	45-64	Samtliga
Arbetare				
Tjänstemän				
därav				
Högre tjänstemän				
Samtliga				

Lägg märke till att sociogrupp-indelningen innehåller "därav-poster", dvs. klasserna är delvis överlappande.

Antag att man är intresserad av följande storheter

$$R_{ij} = \frac{T_{ij}}{N_{ij}} \frac{N_{.j}}{T_{.j}} \quad i, j = 1, 2, 3, 4 \quad (2.1)$$

där $T_{.j} = \sum_i T_{ij}$, dvs för varje kombination av åldersgrupp och sociogrupp vill man skatta kvoten mellan medelinkomsten i cellen och medelinkomsten i hela åldersgruppen.

Antag att data utgörs av ett stratifierat urval med OSU inom strata. Stratumindelningen behöver inte sammanfalla med sociogrupperna eller åldersgrupperna.

Skattningarna av de fyra totaler T_{ij} , N_{ij} , $N_{.j}$ och $T_{.j}$ som bygger upp dubbelkvoten (2.1) är alla parvis korrelerade. Variansskattningen för R_{ij} (som innehåller fyra varianstermer och sex kovarianstermer) kan lätt tas fram med CLAN.

Exempel 2. Antag att tabellen i exempel 1 avser år 1, och att samma urval (panel) används för att ta fram motsvarande tabell år 2. Vissa av individerna i urvalet kan ha bytt sociogrupp och/eller åldersgrupp sedan år 1. Låt $R_{ij}(1)$ och $R_{ij}(2)$ vara dubbelkvoter enligt (2.1) avseende år 1 resp. 2.

Vi vill nu skatta förändringen

$$D_{ij} = R_{ij}(2) - R_{ij}(1) \quad (2.2)$$

och det går utmärkt med CLAN.

Exempel 3. Låt T_{ij} vara en total i cell (i,j) i en tvåvägstabell. Såväl rad- som kolumn-klasserna får vara mer eller mindre överlappande. Antag att kolumn-marginalerna $T_{.j}$ är kända från en totalundersökning eller ett register. I stället för Horvitz-Thompson-skattningen $\hat{T}_{ij}$ kan man använda följande kvotskattning,

$$\hat{T}_{ij}^* = \hat{T}_{ij} \frac{T_{.j}}{\hat{T}_{.j}} \quad (2.3)$$

där rad-marginalerna skattas med

$$\hat{T}_{i.}^* = \sum_j \hat{T}_{ij}^* \quad (2.4)$$

Varianserna för skattningarna i (2.3) och (2.4) kan beräknas med CLAN. (Varianserna för de enskilda cellskattningarna $\hat{T}_{ij}^*$ skulle kunna beräknas med SMED men däremot inte variansen för radmarginalen $\hat{T}_{i.}^*$. Ofta är just radmarginalen den mest intressanta.)

2.2. Designer och parametrar

2.2.1. Urvalsdesigner och bortfallsmodeller

Genom nyckelordet STRATEGI specificerar användaren vilken kombination av urvalsdesign och bortfallsmodell som skall användas. Det finns – liksom i SMED – fyra strategier i CLAN.

En mycket stor del av de urvalsdesigner och bortfallsmodeller som används vid SCB kan på olika sätt återföras på dessa fyra grundstrategier. STRATEGI har exakt samma innebörd i CLAN som i SMED83.

Strategi 1 avser stratifierat urval med OSU utan återläggning. Bortfallsmodellen är oberoende och lika svarssannolikheter inom strata (rak uppräknings).

Strategi 2 Samma urvalsdesign som i strategi 1. Skillnaden ligger i att man här kan dela in strata i svarshomogenitetsgrupper inom vilka objekten har samma svarssannolikheter. Gruppernas storlek antas okända. Det är värt att notera att s.k. tvåfasurval där första fasen består av t.ex. en postenkät till ett stratifierat OSU och andra fasen utgörs av telefonintervjuer av ett subsamplet från första fasens bortfall kan hanteras via STRATEGI 2 (se artikel i Nodus 1986:7).

Strategi 3 används om man har klusterurval, där klustren delats in i strata. Inom varje stratum, väljs klustren med OSU- $U\hat{A}$. Elementen inom de valda klustren totalundersöks. Klustren kan exempelvis vara företag och elementen arbetsställen

inom samma företag. Dessa arbetsställen kan tillhöra olika redovisningsgrupper i en tabell. Under strategi 3 behandlas bortfall av kluster på samma sätt som bortfall av objekt under strategi 1.

Strategi 4 Kombinerar strategi 2 och 3 så att urvalsdesign sammanfaller med strategi 3 och bortfallsmodell med strategi 2.

2.2.2. Övertäckningsmodell

De objekt som ingår i bruttopopulationen men ej i nettopopulationen antas i grundfallet svara i samma utsträckning som objekten i nettopopulationen. I vissa undersökningar vill man dock hellre utgå från antagandet att samtliga övertäckningsobjekt i urvalet kan identifieras, dvs hela bortfallet tillhör nettopopulationen. CLAN är tillämpligt även i detta fall efter en viss modifiering av uppräkningsstalen, se bilaga 1 i Nordberg (1989). Programmet kan naturligtvis också hantera de fall där samtliga övertäckningsobjekt i hela populationen kan identifieras.

2.2.3. Skattningar

Resultatet från CLAN kan ses som en sammansatt tvåvägstabell där varje tabellcell innehåller ett antal par (P, S) av punktskattningar (P) och medelfel (S). Vi börjar med det enklaste fallet där man har *en* tabellcell och *ett* par (P, S).

Låt $X_1, \dots, X_q, \dots, X_Q$ vara ett godtyckligt antal totaler vilka får avse helt godtyckliga redovisningsgrupper $d_1, \dots, d_q, \dots, d_Q$.

Vi uttrycker det matematiskt så att,

$$X_q = \sum_{i=1}^N x_{qi} c_{qi} \quad (2.5)$$

där,

$$c_{qi} = \begin{cases} 1 & \text{om objekt } i \text{ tillhör } d_q \\ 0 & \text{annars} \end{cases}$$

Redovisningsgrupperna får vara godtyckligt överlappande, dvs ett objekt får tillhöra godtyckligt många av grupperna $d_1, \dots, d_q, \dots, d_Q$.

Låt $\hat{X}_1, \dots, \hat{X}_q, \dots, \hat{X}_Q$ vara vanliga (Horvitz–Thompson)–skattningar av totalerna $X_1, \dots, X_q, \dots, X_Q$, utifrån någon av strategierna 1–4, (jfr avsnitt 6).

CLAN kan användas för att skatta medelfelet för

$$\hat{Y} = f(\hat{X}_1, \dots, \hat{X}_q, \dots, \hat{X}_Q) \quad (2.6)$$

där f är en godtycklig rationell funktion. Vi har alltså,

$$P = f(\hat{X}_1, \dots, \hat{X}_q, \dots, \hat{X}_Q) \quad (2.7a)$$

och

$$S = \sqrt{\hat{V} [f(\hat{X}_1, \dots, \hat{X}_q, \dots, \hat{X}_Q)]} \quad (2.7b)$$

CLAN kan i samma körning beräkna (2.7a) och (2.7b) för ett godtyckligt antal funktioner. Utdatasetet kan ha godtyckligt många rader och kolumner och varje cell kan ha godtyckligt många par (P, S). För varje kombination av "rad, kolumn, par" kan man i princip använda olika totaler, olika redovisningsgrupper och olika funktioner f .

SMED kan endast hantera funktionerna, $f = X_1$ och $f = X_1/X_2$ dvs. totaler och kvoter mellan två totaler.

SUDAAN tillåter något fler funktioner än SMED men är mer restriktivt än SMED när det gäller att specificera redovisningsgrupper, och väsentligt mer restriktivt än CLAN både beträffande tillåtna funktioner och specifikation av redovisningsgrupper.

2.3. Jämförelse mellan CLAN och SMED

CLAN är, genom att det är programmerat i SAS, användbart (och ser likadant ut) i såväl PC- som stordatormiljö. SMED kan endast fungera i stordatorn. CLAN kan hantera alla beräkningssituationer som SMED klarar och dessutom åtskilliga andra. Listan med exempel ovan (ex. 1–3) skulle kunna göras mycket längre. Detta innebär dock inte att CLAN i alla avseenden är överlägset SMED. Det finns fall där SMED använder mindre CPU-tid än CLAN. Dessutom är SMED genom den nära kopplingen till TAB 68 bekvämt att använda i TAB 68-baserade produktionssystem.

2.4. Jämförelse mellan CLAN och SUDAAN

CLAN klarar i stort sett alla fall som SUDAAN hanterar. Det viktigaste undantaget gäller tvåstegs klusterurval. Det är dock en design som inte är så vanlig vid SCB. Det finns däremot åtskilliga för SCB viktiga fall som endast CLAN kan hantera. Det bör dock noteras att för vissa enkla designer och tabeller kan SUDAAN ibland vara åtskilligt snabbare än CLAN. En del av förklaringen till detta är att SUDAAN är programmerat i C medan CLAN är programmerat i SAS (som i sin tur ju sedan översätts till C). CLAN kan alltså sägas ligga en språknivå högre än SUDAAN.

En annan del av förklaringen är att CLAN är gjort för att kunna hantera avsevärt mer komplicerade situationer än SUDAAN. Den största skillnaden är att CLAN tillåter helt godtyckliga, överlappande redovisningsgrupper. Ett objekt kan då ingå

i många redovisningsgrupper. SUDAAN förutsätter att redovisningsgrupperna är ömsesidigt uteslutande, och ett objekt kan då endast ingå i en redovisningsgrupp. I vissa fall enkla situationer för vilka SUDAAN skräddarsyts har CLAN visat sig vara avsevärt långsammare än SUDAAN.

3. Användargränssnitt

CLAN beräknar de punkt- och medelfelskattningar som användaren specificerar (inom de ramar som angetts i avsnitt 2). Output består av ett SAS-dataset med namnet DUT som innehåller de begärda skattningarna.

CLAN förutsätter att observerade data ligger på ett SAS-dataset. Datasetet skall, förutom dessa värden, även innehålla variabler för stratumidentitet, populations- och urvals-storlekar etc som användaren refererar till i ett SAS-macro *%VARIANS* (se nedan).

Användaren måste skriva ett macro *%FUNKTION* som specificerar vilka funktioner av totaler CLAN skall beräkna punktskattningar och medelfel för. *%FUNKTION* anropas av *%VARIANS* och måste alltså komma före *%VARIANS* i användarens program.

Huvudstrukturen i en CLAN-applikation ser ut på följande sätt,

```
%INCLUDE 'CLAN';      /* Till exempel */
  [DATA;]
  .
  .
  .
%MACRO FUNKTION(.,.);
  .
  .
%MEND;
%VARIANS(.,., .)
```

DATA-steget är inte nödvändigt om det redan finns ett färdigt dataset med nödvändig information.

Eftersom resultatet från CLAN utgörs av ett SAS-dataset med skattningar kan användaren vidarebearbeta resultaten efter egna önskemål. Ett enkelt sätt att se resultatet är att avsluta en CLAN-applikation med `PROC PRINT;` som ger en utskrift av innehållet i resultatfilen DUT, på bildskärmen eller på papper. Man kan också ta hand om DUT på annat sätt, t ex genom att använda `PROC TABULATE`. Mer om detta i avsnitt 4.

Nedan följer en detaljerad presentation av användargränssnittets olika delar.

%VARIANS (DATA=, STRATEGI=, STRATUM=, GRUPP=, NSVAR=,
NPOP=, NGRUPP=, NURV=, ID=, MAXRAD=, MAXKOL=)

Parametrar som alltid **måste** finnas,

NSVAR	anger vilken variabel som innehåller antal "svar" per stratum, inklusive övertäckningen, m_h för STRATEGI 1 och 3, m_{hg} för STRATEGI 2 och 4.
NPOP	anger vilken variabel som innehåller antal objekt i urvalsramen i respektive stratum, N_h .
MAXRAD	antalet rader vid indelningen av populationen i redovisningsgrupper, (se nedan).
MAXKOL	antalet kolumner vid indelningen av populationen i redovisningsgrupper, (se nedan).

Parametrar som alltid **kan** anges,

[DATA]	anger SAS-datasetets namn, om det utelämnas används det sist skapade datasetet.
[STRATEGI]	anger vilken strategi som skall tillämpas och styr även vilka övriga argument som skall anges, kan anta värdena 1, 2, 3, 4. Om STRATEGI saknas eller är blank sätts STRATEGI=1.
[STRATUM]	anger vilken variabel som innehåller stratumidentiteten. Om STRATUM saknas kommer OSU att förutsättas. Datasetet förutsätts vara sorterat efter stratum.

Parametrar som **måste** finnas för STRATEGI 2 och 4.

GRUPP	anger vilken variabel som innehåller gruppidentiteten. Datasetet förutsätts sorterat efter STRATUM och inom stratum efter GRUPP.
NGRUPP	anger vilken variabel som innehåller antalet utvalda objekt i GRUPP, n_{hg} .
NURV	anger vilken variabel som innehåller antalet utvalda objekt i STRATUM, n_h .

Parametrar som **måste** finnas för STRATEGI 3 och 4,

ID	anger vilken variabel som innehåller klusteridentiteten. Datasetet förutsätts sorterat efter STRATUM och ID om STRATEGI=3 och efter STRATUM, GRUPP och ID om STRATEGI=4.
----	--

I många fall vill man kunna gruppera sina skattningar efter nivåerna på en eller två faktorer. DUT innehåller därför två hjälp-variabler "RAD" och "KOL" som är till för att hjälpa användaren att strukturera DUT.

RAD antar värdena 1, 2, ..., MAXRAD och KOL värdena 1, 2, ..., MAXKOL.

Användaren bestämmer antalet nivåer i respektive gruppering och därmed antalet poster i DUT genom att ange MAXRAD och MAXKOL i %VARIANS, antalet poster blir alltid MAXRAD x MAXKOL. Variablerna RAD och KOL används t ex för att underlätta identifieringen av de olika skattningarna.

Antag att vi för samtliga 16 kombinationer av socialgrupp och ålder i Exempel 1 ovan vill ha punktskattningar och medelfel för T_{ij} , N_{ij} och $R_{ij}=T_{ij}/N_{ij}$. Vi vill alltså beräkna 48 ($4 \times 4 \times 3$) par (P, S) av punktskattningar (P) och medelfel (S).

Skattningarna kan läggas ut i DUT på olika sätt. Det naturligaste sättet är kanske följande,

social- grupp	ålders- grupp	PT	ST	PN	SN	PR	SR
1	1						
.	2						
.	3						
1	4						
.	.						
4	1						
.	2						
.	3						
4	4						

I detta fall är MAXRAD=4 och MAXKOL=4. Resultatet i DUT består nu av 16 poster genererade av 4 nivåer av en "radindikator" (socialgrupp) och 4 nivåer av en "kolumnindikator" (ålder). Varje post innehåller variablerna RAD och KOL samt tre par variabler (P_x , S_x), dvs 8 SAS-variabler och det är lätt att koppla ihop detta med tabellen i exempel 1.

Det är dock inget som hindrar att man i stället lägger ut samtliga skattningar som *en* post i DUT med de 48 paren som $2 \times 48 = 96$ SAS-variabler. Ordna tabellcellerna i exempel 1 i godtycklig ordning. Bilda variablerna T1, T2, ..., T16, N1, ..., N16, R1, ..., R16, där exempelvis T5 betyder totalinkomst i cell nummer 5 osv. DUT får då utseendet,

RAD	KOL	PT1	ST1	.	.	.	PR16	SR16
1	1							

Det är fullt möjligt att ta ut DUT på detta sätt genom att sätta MAXRAD=1 och MAXKOL=1 men det kan vara svårt att göra kopplingen till tabellen i exempel 1.

I *%VARIANS* specificeras alltså de statistiska förutsättningarna genom valet av strategi etc. Dessutom bestäms antalet poster i DUT genom MAXRAD och MAXKOL.

Användaren måste också specificera vilka skattningar som skall finnas på respektive post, (kombination av RAD och KOL), i DUT. Detta görs genom att användaren skriver en *%macro FUNKTION (arg1, arg2)*; med syfte att specificera vilka punkt- och medelfelskattningar han/hon vill ha. Argumenten *arg1*, *arg2* är formella parametrar som anger värde på RAD (*=arg1*) och KOL (*=arg2*), dvs *arg1* antar värdena 1, 2, ..., MAXRAD, och *arg2* antar värdena 1, 2, ..., MAXKOL. Argumenten får ha vilka giltiga namn på SAS-variabler som helst, dock begränsade till 6 tecken. Undvik namn som börjar med *_* (understrykning).

Till sin hjälp har användaren följande färdiga macros,

<i>%TOT</i>	anropas för varje total som skall skattas och/eller ingå i en funktion.
<i>%ADD</i>	summerar en (=tilldelning) eller två skattningar av totaler eller funktioner av totaler.
<i>%SUB</i>	beräknar skillnaden mellan två skattningar av totaler eller funktioner av totaler.
<i>%DIV</i>	beräknar kvoten mellan två skattningar av totaler eller funktioner av totaler
<i>%MULT</i>	beräknar produkten av två skattningar av totaler eller funktioner av totaler.
<i>%ESTIM</i>	anger vilka skattningar av totaler eller funktioner av totaler med medelfelskattningar som skall läggas ut på resultatfilen, <i>%ESTIM</i> tillåter även att man döper om variabeln på resultatfilen.

Gemensamt för alla *%macron* är att argumentet *namn* förekommer, *namn* får ersättas med valfritt SAS-namn, dock begränsat till 6 teckens längd. Namnen bör inte börja med *_* (understrykning).

%TOT (namn, variabel, villkor)

Här definierar användaren vilken total han/hon vill skatta och /eller använda för sina funktioner. Totalerna **måste** definieras, dvs minst en *%TOT* **måste** finnas.

Argumentet *namn* används för att senare kunna referera till rätt total, *variabel* är namn på den variabel i det dataset som DATA refererar till i *%VARIANS*, och som totalen skall skattas för.

Villkor kan bestå av antingen en variabel eller ett villkorsuttryck som har värdet =1 om observationen skall ingå. Genom att referera till *arg1* och *arg2* i *%macro FUNKTION*, kan man i villkoret styra hur respektive redovisningsgrupp skall definieras.

%ADD(*namn*, *fkn1*, [*fkn2*])

Här anger användaren att variabeln *namn* skall vara lika med *fkn1*[+*fkn2*], om *fkn2* saknas, tilldelas *namn* värdet av *fkn1*.

%SUB (*namn*, *fkn1*, *fkn2*)

Här anger användaren att variabeln *namn* skall vara lika med *fkn1*−*fkn2*.

%DIV (*namn*, *fkn1*, *fkn2*)

Här anger användaren att variabeln *namn* skall vara lika med *fkn1*/*fkn2*. Om *fkn2*=0 blir resultatet =0.

%MULT(*namn*, *fkn1*, *fkn2*)

Här anger användaren att variabeln *namn* skall vara lika med *fkn1***fkn2*.

Samma *namn* får användas flera gånger i samtliga macros ovan, t.ex. %DIV (SUM, ETT, TU) och %ADD(SUM, SUM, TRE), i nu nämnd ordning, resulterar i att SUM=ETT/TU+TRE.

%ESTIM (*namn*, [*utnamn*])

I %ESTIM anger *namn* vilken funktion som skall redovisas med sin punkt-skattning och medelfelskattning. Punkt- och medelfel-skattningarna får variabelnamnen *Putnamn* och *Sutnamn*.

Om *utnamn* utelämnas får skattningarna variabelnamnen *Pnamn* och *Snamn*. På resultatfilen DUT kommer dessutom SAS-variablerna RAD och KOL att finnas som identifierar rader och kolumner. Variablerna RAD och KOL antal värdena 1, 2, ..., MAXRAD, respektive 1, 2,..., MAXKOL. Minst en %ESTIM måste finnas, annars kommer inga skattningar ut.

Exempel

%ESTIM (Z) ger namnen PZ och SZ medan

%ESTIM (Z, ANDEL) ger namnen PANDEL och SANDEL för samma skattningar.

I %macro FUNKTION definierar användaren de funktioner han/hon vill ha och hur de skall redovisas. Macrona %TOT, %ADD, %SUB, %DIV, %MULT får förekomma i valfritt antal men argumenten *fkn1* och *fkn2* till dem måste vara definierade genom anrop av %TOT innan de används. %TOT får bara referera till "elementära" variabler som redan existerar på SAS-filen, dvs *variabeln* får inte vara definierad av %ADD, %SUB, %DIV eller %MULT.

%macro FUNKTION (*arg1*, *arg2*) får innehålla även andra data-satser, t.ex. ARRAY, DO, IF-THEN-ELSE etc.

%macro FUNKTION (arg1, arg2) skall definieras före *%VARLANS* .

4. Hur gör man?

4.1. Allmänt

När man skriver den *%macro FUNKTION(R,K)*; som krävs för att man skall få de punktskattningar med skattningar av medelfel som man önskar, bör man ha i åtanke att *FUNKTION* ligger i en loop i CLAN.

CLAN fungerar alltså så att för varje observation som läses från den datafil som CLAN skall arbeta med, kontrolleras om observationen tillhör "cell" i,j, om så är fallet ger observationen ett bidrag till skattningen enligt *%FUNKTION(R,K)*;

Detta betyder att internt i CLAN finns en sekvens,

```
.  
.
DO I=1 TO &MAXRAD;
DO J=1 TO &MAXKOL;
  %FUNKTION( I, J )
END;
END;
.  
.
```

Det är viktigt att *%macro FUNKTION* kommer före *%VARLANS*.

Användaren kan testa på &R resp &K i *%macro FUNKTION(R,K)*, för att exakt styra vilka beräkningar som skall ske i vilka celler eller för att kombinera skattningar i olika celler.

(I SAS macro-språk är R och K i exemplet ovan formella parametrar. För att referera till deras värden används &R resp &K.)

CLAN behandlar *%macro FUNKTION(R,K)* i tre faser, dvs CLAN refererar till *FUNKTION* vid tre tillfällen. Samtliga hjälpfunktioner *%TOT*, *%ADD*, ... , *%ESTIM* fungerar på olika sätt i olika faser. Internt används en macrovariabel *ind* för att styra vad funktionerna skall göra, även denna kan användaren testa på.

Variabeln *ind* antar i tur och ordning, värdena 0, 1, 2, under exekveringen.

&ind=0 innebär viss syntaxkontroll samt beräkning av minnesutrymme för hjälpfunktionerna. I denna fas finns inga referenser till datafilen vilket kan medföra felutskriften av typen "Variable V1 is uninitialized". Sådana fel i denna fas har dock ingen inverkan på den fortsatta bearbetningen.

&ind=1 betyder att enbart skattningar av totalerna beräknas, inga funktioner av totaler beräknas, dvs *%ADD*, ..., *%ESTIM* refereras inte, inga medelfelsberäkningar görs.

&ind=2 betyder att alla funktioner i *%macro FUNKTION(R,K)*, beräknas, punktskattningar och medelfel beräknas för alla funktioner som angivits i *%ESTIM*, en resultatfil *DUT* skapas.

En varning är dock på sin plats, **ändra ALDRIG macrovariablerna *MAXRAD*, *MAXKOL*, *R*, *K* (motsv) eller *ind* i *%FUNKTION* !**

4.2. Exempel

4.2.1. STRATEGI=1

I avsnitt 2 visades tre exempel på skattningar som kan utföras med CLAN.

I **Exempel 1** vill man skatta,

$$R_1 = \frac{T_{ij}}{N_{ij}} \frac{N_{.j}}{T_{.j}}, \quad i,j=1, 2, 3, 4,$$

där T_{ij} och N_{ij} är totalinkomst respektive antal personer i cell (i,j) , där i står för Socialgrupp och j för åldersgrupp samt $i=4$ och $j=4$ svarar mot "Samtliga".

Data har erhållits via ett stratifierat urval med OSU inom strata, bortfallskompensation sker genom "rak" uppräknings, totalerna skattas med,

$$\hat{T}_{ij} = \sum_{h=1}^H \frac{N_h}{m_h} \sum_{k=1}^{m_h} c_{ijk} y_{hk}, \quad \text{där } c_{ijk}=1 \text{ om observation } k \text{ tillhör } hij, 0 \text{ annars.}$$

respektive,

$$\hat{N}_{ij} = \sum_{h=1}^H \frac{N_h}{m_h} \sum_{k=1}^{m_h} c_{ijk}.$$

$T_{.j}$ och $N_{.j}$ skattas på analogt vis. Kvoten R_1 skattas med,

$$\hat{R}_1 = \frac{\hat{T}_{ij}}{\hat{N}_{ij}} \frac{\hat{N}_{.j}}{\hat{T}_{.j}}$$

Data finns i SAS-filen INKDATA som bland annat innehåller variablerna STR (stratumidentitet), POP (N_h), SVAR (m_h), ARB1 (=1 om arbetare, 0 annars), LTJ1 (=1 om lägre tjm, 0 annars), HTJ1 (=1 om högre tjm, 0 annars), ALD1 (åldersklass 1, 2, 3), INK1 (inkomst).

Filen antages sorterad i fallande eller stigande sekvens efter STR.

Följande FUNKTION ger en skattning av R_1 och dess medelfel, observera att eftersom argumenten till %FUNKTION är formella parametrar står det oss fritt att använda vilka SAS-variabelnamn som helst för att identifiera rader och kolumner, här används R och C,

```
%MACRO FUNKTION(R,C);

%*   RV1 och KV1 är >0 för de celler som en obser-
      vation bidrar till (RV1 och KV1 betecknar rad- och
      kolumnvillkor). Satserna nedan kommer att generera
      felutskrifter, "Variable ARB1 is uninitialized"
      etc, dessa felutskrifter kan ignoreras, se 4.1
      ovan, de kan också undvikas, se nedan.;

IF &R=1 THEN RV1=ARB1;
IF &R=2 THEN RV1=LTJ1+HTJ1;
IF &R=3 THEN RV1=HTJ1;
IF &R=4 THEN RV1=1;
IF 1<=&C<=3 THEN KV1=(&C=ALD1);
IF &C=4 THEN KV1=1;

%*   Totalerna skattas;

%TOT (T1ij,INK1,RV1 and KV1)
%TOT (N1ij,1,RV1 and KV1)
%TOT (T1_j,INK1,KV1)
%TOT (N1_j,1,KV1)

%*   Kvoterna bildas;

%DIV (Q1ij,T1ij,N1ij)
%DIV (Q1_j,T1_j,N1_j)
%DIV (R1,Q1ij,Q1_j)

%*   Medelfelen beräknas och en resultatfil, DUT,
      med variablerna PR1 och SR1 genereras;

%ESTIM (R1)

%*macrot avslutas;

%MEND;
```

Detta är bara *ett* av många sätt att skriva FUNKTION på. Kvoten R_1 kan bildas t ex med,

```
%MULT (Q1ij,T1ij,N1_j)
%DIV (Q1ij,Q1ij,N1ij)
%DIV (R1,Q1ij,T1_j)
```

vilket ger samma resultat.

För att starta alltihop skriver man,

```
%VARIANS (DATA=INKDATA, STRATUM=STR, NPOP=POP,
          NSVAR=SVAR, MAXRAD=4, MAXKOL=4)
```

För att undvika felutskrifterna som beror på de 6 första raderna kan man i *%macro FUNKTION*, skriva t ex,

```
%if &ind >0 %then %do;
    IF &R=1 THEN RV1=ARB1;
    IF &R=2 THEN RV1=LTJ1+HTJ1;
    IF &R=3 THEN RV1=HTJ1;
    IF &R=4 THEN RV1=1;
    IF 1<=&C<=3 THEN KV1=(&C=ALD1);
    IF &C=4 THEN KV1=1;
%end;
```

I **Exempel 2** antas att samma skattningar görs vid tidpunkt 2 för samma individer som i Exempel 1. Vid tidpunkt 2 kommer en del individer att ha bytt åldersklass och/eller socialgrupp. Vi är intresserade av kvoten R_2 som definieras analogt med R_1 ovan, dessutom är vi intresserade av differensen mellan kvoterna, $R_2 - R_1$.

Kvoten R_2 skattas analogt med $\hat{R}_1$ och differensen $D_{21} = R_2 - R_1$ skattas med $\hat{D}_{21} = \hat{R}_2 - \hat{R}_1$.

På filen INKDATA antas variablerna ARB2, LTJ2, HTJ2, ALD2 och INK2 finnas, förutom de variabler som angavs i Exempel 1. *%macro FUNKTION* ser ut på analogt sätt som i Exempel 1,

```
%MACRO FUNKTION(R,C);
    IF &R=1 THEN RV1=ARB1;
    IF &R=2 THEN RV1=LTJ1+HTJ1;
    IF &R=3 THEN RV1=HTJ1;
    IF &R=4 THEN RV1=1;
    IF 1<=&C<=3 THEN KV1=(&C=ALD1);
    IF &C=4 THEN KV1=1;

    %TOT (T1ij,INK1,RV1 and KV1)
    %TOT (N1ij,1,RV1 and KV1)
    %TOT (T1_j,INK1,KV1)
    %TOT (N1_j,1,KV1)

    %DIV (Q1ij,T1ij,N1ij)
    %DIV (Q1_j,T1_j,N1_j)
    %DIV (R1,Q1ij,Q1_j)
```

```
IF &R=1 THEN RV2=ARB2;
IF &R=2 THEN RV2=LTJ2+HTJ2;
IF &R=3 THEN RV2=HTJ2;
IF &R=4 THEN RV2=1;
IF 1<=&C<=3 THEN KV2=( &C=ALD2);
IF &C=4 THEN KV2=1;

%TOT (T2ij, INK2, RV2 and KV2)
%TOT (N2ij, 1, RV2 and KV2)
%TOT (T2_j, INK1, KV2)
%TOT (N2_j, 1, KV2)

%DIV (Q2ij, T2ij, N2ij)
%DIV (Q2_j, T2_j, N2_j)
%DIV (R2, Q2ij, Q2_j)

%*   Bilda differensen;

%SUB (D21, R2, R1)

%ESTIM (R1)
%ESTIM (R2)
%ESTIM (D21)

%MEND;
```

På resultatfilen DUT kommer att finnas variablerna PR1, SR1, PR2, SR2, PD21, SD21.

I **Exempel 3** antogs att marginaltotalerna $T_{.j}$ var kända, t ex från en totalundersökning eller ett register. Genom att utnyttja dessa storheter kan T_{ij} och $T_{i.}$ skattas effektivare än om estimatorerna i Exempel 1 används. En estimator för T_{ij} är t

ex $\hat{T}_{ij}^* = T_{.j} \frac{\hat{T}_{ij}}{\hat{T}_{.j}}$ och en estimator för $T_{i.}$ är t ex $\hat{T}_{i.}^* = \sum_{j=1}^4 \hat{T}_{ij}^*$. Kolumnerna kanske

inte är intressanta i sig men används för att få effektiva skattningar av radtotalerna. Infilen antages innehålla samma variabler som i Exempel 1 och dessutom finns variabeln TINK1 som innehåller den kända totalen för varje åldersgrupp. I CLAN kan man skriva %macro FUNKTION (R,K) på följande sätt,

```
%MACRO FUNKTION(R,C);
IF &R=1 THEN RV1=ARB1;
IF &R=2 THEN RV1=LTJ1+HTJ1;
IF &R=3 THEN RV1=HTJ1;
IF &R=4 THEN RV1=1;
IF 1<=&C<=3 THEN KV1=( &C=ALD1);
IF &C=4 THEN KV1=1;

IF &C < &MAXKOL THEN DO;
    %TOT (T1ij, INK1*TINK1, RV1 and KV1)
    %TOT (T1_j, INK1, KV1)

    %DIV (Q1, T1ij, T1_j)
```

```
IF &C=1 THEN %ADD(TS,Q1)
IF &C>1 THEN %ADD(TS,TS,Q1)
END;

ELSE %ADD(Q1,TS)

%ESTIM (Q1,POST)

%MEND;
```

På resultatfilen DUT finns variablerna PPOST och SPOST.

4.2.2. STRATEGI=2

I Exempel 1–3 förutsättes STRATEGI=1. För det fall man vill använda STRATEGI=2, för att t ex kompensera skattningarna för bortfallseffekter genom gruppvis uppräknings eller genom att använda svarshomogenitetsgrupper, krävs ytterligare information på infilen. Varje stratum delas in i ett antal uppräkningsgrupper, för varje grupp finns en variabel som anger grupptillhörighet, en variabel som innehåller antalet utvalda objekt i stratum, en variabel som anger antalet utvalda objekt i gruppen, en variabel som anger antalet svar i gruppen, låt oss kalla dessa GRP, LILLAN, NGRP, MGRP respektive.

SAS-filen förutsättes sorterad efter stratumtillhörighet och inom strata, efter grupptillhörighet.

FUNKTION påverkas inte av vilken STRATEGI man väljer utan det anger man i %VARIANS. För alla tre exemplen kan %VARIANS anges på följande sätt om STRATEGI=2 är tillämpligt,

```
%VARIANS (DATA=INKDATA, STRATEGI=2, STRATUM=STR,
          NPOP=POP, NURV=LILLAN, NGRUPP=NGRP,
          GRUPP=GRP, NSVAR=MGRP, MAXRAD=4,
          MAXKOL=4)
```

4.2.3. STRATEGI=3

På samma sätt som i fallet med STRATEGI=2 är FUNKTION och urvalsdesign oberoende av varandra, dvs om designen är enstegs klusterurval där klustren eller de primära urvalsenheterna, PUEn, stratifierats före urvalsdragningen och urvalet av PUEn sker med OSU, kan samma FUNKTION användas som i Exempel 1–3.

Vi illustrerar STRATEGI 3 med ett exempel, Exempel 4, från jordbrukstatistiken. Lantbruksföretagen i LantBruksRegistret, LBR, har stratifierats. I varje stratum har ett OSU av lantbruksföretag dragits. Ett lantbruksföretag kan ha flera brukare, för varje brukare frågas bland annat efter ålder och arbetad tid i djurhållning och i växtodling. Vi vill skatta arbetad tid i djurhållning och växtodling, fördelat på riksområde, RO1–3 och ålder hos brukaren, 8 åldersklasser enligt nedan,

	Ålder 1	Ålder 2	...	...	...	...	Ålder 8
RO 1							
RO 2							
RO 3							

Varje post (=en brukare) på SAS-filen JAK innehåller stratumidentitet, STR, antalet lantbruksföretag i LBR, POP, företagsidentitet, LBRNR och antal svarande företag, SVAR . Dessutom finns variablerna AGRUPP för åldersgrupp, RO för företagets regiontillhörighet, DJUR för brukarens arbetade tid i djurhållning och VAXT för brukarens arbetade tid i växtodling.

Data förutsätts sorterat efter stratumidentitet och inom stratum efter företagsidentitet, t ex med,

```
PROC SORT DATA=JAK; BY STR LBRNR;
```

I CLAN får man önskade punkt- och medelfelskattningar på följande sätt,

```
%MACRO FUNKTION (R,C);  
%TOT(DJUR,DJUR,&R=RO and &C=AGRUPP)  
%TOT(VAXT,VAXT,&R=RO and &C=AGRUPP)  
  
%ESTIM(DJUR)  
%ESTIM(VAXT)  
  
%MEND;  
  
%VARIANS (DATA=JAK,STRATUM=STR,STRATEGI=3,  
          NSVAR=SVAR,ID=LBRNR,NPOP=POP,MAXRAD=3,  
          MAXKOL=8)
```

4.3. Tabeller

CLAN producerar en SAS-fil, DUT. Denna kan användas för tabellframställning, t ex med hjälp av PROC TABULATE eller något annat programpaket. Resultatet från Exempel 3 kan t ex tabelleras på följande sätt,

```
PROC TABULATE DATA=DUT;  
  VAR PPOST SPOST;  
  CLASS RAD KOL;  
  TABLE RAD*(( PPOST SPOST)*F=BEST. ), KOL;
```

Resultatet blir en tabell med 4 kolumner och 8 rader eftersom punktskattning och medelfel hamnar under varandra.

Om användaren vill ha t ex relativa medelfelet i sina tabeller, kan detta enkelt erhållas genom att utnyttja ett DATA-steg i SAS där lämpliga transformationer görs, t ex,

```
IF PPOST NE 0 THEN SPOST=SPOST/PPOST;  
ELSE SPOST=. ;
```

5. Kapacitet prestanda och felmeddelanden

Det finns inga inbyggda begränsningar i CLAN. I princip kan hur stora tabeller och hur komplicerade funktioner som helst hanteras. I praktiken sätter dock maskinkapaciteten (och möjligen begränsningar i SAS) gränser för vad som är möjligt. Minneskapaciteten bestämmer hur stora och hur många tabeller som kan genereras. Diskutrymmet sätter gränser för hur många observationer och/eller variabler som kan hanteras på en gång. Användarens tålamod och maskinens snabbhet sätter också gränser för vad som är möjligt.

CLAN är skrivet i SAS macrospråk, SAS är **inte** ett system för att generera program som exekveras snabbt i maskinen. Nedan ges några exempel på körningar och hur lång tid de tagit, både under PC-DOS, OS/2 och SCBs stordator (sept 1990).

Testen under PC-DOS har gjorts på en IBM PS/2 55 SX med matteprocessor, 4MB minne, DOS 3.3 och under OS/2 på en IBM PS/2 mod 70 med matteprocessor, 8MB minne.

5.1. Test 1

Material 1 består av 5280 observationer fördelade på 199 strata. Fyra variabler har använts, AL, BL, AU, BU. Tio redovisningsgrupper vilka kan vara varandra överlappande, har bildats. För varje variabel har totalen skattats i var och en av redovisningsgrupperna.

Kvoten mellan totalen för variabel AL i var och en av de tio grupperna och totalen för AL i grupp 1 har skattats. På samma sätt har kvoten mellan totalen för BL och totalen för BL i grupp 1 skattats.

Totalt skall 60 punktskattningar och 60 medelfelskattningar beräknas. Vi har jämfört CLAN, SUDAAN och SMED83. CLAN och SMED83 jämfördes på stordator, medan CLAN och SUDAAN jämfördes på PC under DOS.

På stordatorn låg data lagrade så att en SAS-fil måste bildas för CLAN. Eftersom data innehöll såväl decimalpunkter som minustecken måste data transformeras till en form som SMED83 kunde acceptera, detta gjordes i ett SAS-datasteg.

Resultat 1

	PC-DOS	Stordator
CLAN	20 min	11.9 s *)
SUDAAN	2 min 20 s	-
SMED83	-	11.8 s **)

*) DATA-steg 3 s

**) DATA-steg 6 s

Ett fall där vi dessutom bildade differensen mellan kvoterna testades också, detta fall kunde bara CLAN klara av. Tiden på stordator ökade med 0.3 s, och tiden under PC-DOS ökade med drygt 1 min.

5.2. Test 2

Material 2 består av 2500 observationer fördelade på tre strata. Materialet har körts på stordator, under PC-DOS och OS/2.

En variabel har använts, totalen för denna har skattats i en 5x5 tabell, dvs 25 punktskattningar och 25 medelfelskattningar har beräknats.

Resultat 2

	PC-DOS	Stordator	OS/2
CLAN	8 min 43 s	5.2 s *)	4 min 50 s
SUDAAN	1 min	-	-
SMED83	-	3.8 s **)	-

*) DATA-steg 1.4 s

**) DATA-steg 1.4 s

Samma data användes för att beräkna skattningarna enligt Exempel 3 i avsnitt 4.2 ovan. Dessa skattningar kan bara erhållas med CLAN, tiderna blev,

PC-DOS	Stordator	OS/2
8 min 20 s	7.2 s	3 min 55 s

5.3. Test 3

Materialet till test 3 bestod av 5837 observationer fördelade på 200 strata. Fyra variabler användes och 100 (10x10) redovisningsgrupper definierades. I varje redovisningsgrupp skattades 4 totaler, 2 kvoter och 1 differens mellan kvoter, dvs 700 punktskattningar och 700 medelfelskattningar beräknades. Detta tog 2 tim 49 min under PC-DOS. I stordatorn tog detta 58 s. Endast CLAN kördes här eftersom varken SMED83 eller SUDAAN klarade detta fall.

5.4. Felutskrifter

CLAN förlitar sig i stor utsträckning på SAS egna felutskrifter, för att kunna se hela program texten som SAS refererar till måste användaren ange `OPTIONS MPRINT;` vilket ger utskrift av all programkod som genereras av de olika macrona.

Det finns dock ett antal varningar och felutskrifter som genereras av CLAN, dessa felutskrifter kommer på loggen och föregås alltid av `*****`. Den enda syntaxkontroll som CLAN gör är att kontrollera att nödvändiga variabler har angivits för de olika strategierna. Om CLAN upptäcker ett allvarligt fel ges kommandot `ABORT;` .

Dessutom finns en "dummy" `%macro FUNKTION` som används om inte användaren har skrivit någon egen `FUNKTION` eller lagt den efter `%VARIANS`. Denna "dummy"-macro upplyser bara om att `FUNKTION` saknas eller ligger i fel ordning samt avbryter körningen med kommandot `ABORT;`.

6. Formelapparaten

6.1. Allmänt

CLAN kan enligt tidigare beskrivning användas för att skatta variansen för en godtycklig rationell funktion av totaler. Antag att parametern man är intresserad av kan uttryckas som $Y=f(X)$, där $X=(X_1, \dots, X_q, \dots, X_Q)$ är olika totaler i populationen U .

Parametern Y skattas med,

$$\hat{Y}=f(\hat{X}_1, \dots, \hat{X}_q, \dots, \hat{X}_Q) \quad (6.1)$$

Totalskattningarna $\hat{X}_q$ antas vara linjära skattningar, exempelvis "vanliga" Horvitz-Thompson skattningar av typen,

$$\hat{X}_q = \sum_i w_i x_{qi} \quad (6.2)$$

De formler som används för att skatta variansen för $\hat{Y}$ bygger på linjärisering av funktionen f .

Låt $f'_q(X)$ vara första partiella derivatan av f med avseende på X_q i punkten X . Då gäller approximativt,

$$V(\hat{Y}) = \sum_q \sum_p f'_q(X) f'_p(X) C(\hat{X}_q, \hat{X}_p) \quad (6.3)$$

Observera att $C(\hat{X}_q, \hat{X}_q) = V(\hat{X}_q)$. En skattning av $V(\hat{Y})$ erhålles genom att skatta kovarianstermerna $C(\cdot, \cdot)$ från samplet och ersätta X med $\hat{X}$ i $f'_q(X)$.

Det är lätt att se att variansformeln (6.3) kan innehålla åtskilliga termer. Om antalet ingående totaler i (6.1) är Q kommer antalet termer i (6.3) att bli $Q(Q+1)/2$. Om Q är stort och samtliga totaler får avse helt godtyckliga och sinsemellan olika redovisningsgrupper som dessutom kan vara delvis överlappande samtidigt som ett stort antal funktioner skall skattas så är det lätt att inse att man får mycket stora problem att organisera och administrera variansberäkningarna.

Problemet har lösts så att CLAN utnyttjar en mycket användbar transformation som gör att problemet att skatta variansen för $\hat{Y}$ kan överföras till att skatta variansen för skattningen av en total. Man slipper därmed bland annat hålla reda på och beräkna kovarianstermerna i (6.3) ovan.

6.2. En användbar transformation

Gör följande transformation, se Woodruff (1971),

$$z_i = \sum_k f'_q(\hat{X}) x_{qi} \quad (6.4)$$

och låt,

$$\hat{Z} = \sum_i w_i z_i \quad (6.5)$$

där w_i är samma vikter som i (6.2).

Om man ser z_i som "nya" observationsdata och skattar variansen för $\hat{Z}$ får man under allmänna villkor exakt samma resultat som om man skattar högerledet i (6.3).

Man har därmed överfört problemet att skatta variansen $\hat{Y}$ till att skatta variansen för en total vilket är ett väsentligt enklare problem än det ursprungliga. Priset man får betala är att man först måste ta fram punktskattningarna $\hat{X}$ och beräkna $f'_q(\hat{X})$. Det är därför nödvändigt att löpa igenom datafilen två gånger, första gången för att bestämma $\hat{X}$ och andra gången för att göra transformationen (6.4) och skatta variansen för $\hat{Z}$. Det bör noteras att det bara är variansen för $\hat{Y}$ som skattas via (6.4), punktskattningen $\hat{Y}$ tas fram direkt via (6.1).

Vi ger här Woodruff-transformationen i två konkreta exempel.

Kvot mellan två totaler

$$\text{Sätt } \hat{Y} = \frac{\hat{X}_1}{\hat{X}_2}$$

Derivering och litet algebra ger

$$z_i = \frac{\hat{X}_1}{\hat{X}_2} \left(\frac{x_{1i}}{\hat{X}_1} - \frac{x_{2i}}{\hat{X}_2} \right) \quad (6.6)$$

Produkt mellan två totaler

$$\text{Sätt } \hat{Y} = \hat{X}_1 \cdot \hat{X}_2$$

Derivering och litet algebra ger

$$z_i = \hat{X}_1 \cdot \hat{X}_2 \left(\frac{x_{1i}}{\hat{X}_1} + \frac{x_{2i}}{\hat{X}_2} \right) \quad (6.7)$$

6.3. Stegvis användning av Woodruff-transformationen för sammansatta funktioner

Vi börjar med ett exempel. Antag att man vill skatta variansen för

$$\hat{Y} = \frac{\hat{X}_1 \cdot \hat{X}_2}{\hat{X}_3 \cdot \hat{X}_4} \quad (6.8)$$

Vi tillämpar (6.4) för att ta fram z . Derivering och litet algebra ger,

$$z_i = \frac{\hat{X}_1 \cdot \hat{X}_2}{\hat{X}_3 \cdot \hat{X}_4} \left(\frac{x_{1i}}{\hat{X}_1} + \frac{x_{2i}}{\hat{X}_2} - \frac{x_{3i}}{\hat{X}_3} - \frac{x_{4i}}{\hat{X}_4} \right) \quad (6.9)$$

Vi visar nu att man kan komma till (6.9) genom att ta transformationen i två steg.

$$\text{Sätt } \hat{Y}_1 = \frac{\hat{X}_1}{\hat{X}_3},$$

$$\hat{Y}_2 = \frac{\hat{X}_2}{\hat{X}_4} \text{ och}$$

$$\hat{Y} = \hat{Y}_1 \cdot \hat{Y}_2.$$

Transformera för $\hat{Y}_1$ och $\hat{Y}_2$. Analogt med (6.6) erhålles

$$z_{1i} = \frac{\hat{X}_1}{\hat{X}_3} \left(\frac{x_{1i}}{\hat{X}_1} - \frac{x_{3i}}{\hat{X}_3} \right) \text{ och } z_{2i} = \frac{\hat{X}_2}{\hat{X}_4} \left(\frac{x_{2i}}{\hat{X}_2} - \frac{x_{4i}}{\hat{X}_4} \right) \quad (6.10)$$

Insättning av (6.10) i (6.9) ger,

$$z_i = \hat{Y}_1 \cdot \hat{Y}_2 \left(\frac{z_{1i}}{\hat{Y}_1} + \frac{z_{2i}}{\hat{Y}_2} \right) \quad (6.11)$$

Notera att (6.11) är den transformation som svarar mot produkten $\hat{Y}_1 \cdot \hat{Y}_2$ med de transformerade värdena z_{1i} och z_{2i} som indata, jämför (6.7).

Det skulle också varit möjligt att göra transformationen på ett annat sätt, t ex genom $\hat{Y}_1 = \hat{X}_1 \cdot \hat{X}_2$, $\hat{Y}_2 = \hat{X}_3 \cdot \hat{X}_4$ och $\hat{Y} = \hat{Y}_1 / \hat{Y}_2$, vilket leder till samma resultat.

Det är fullt möjligt att lösa upp transformationen i ytterligare steg om man har en mer komplicerad funktion än (6.8).

En rationell funktion kan åstadkommas genom upprepad användning av de fyra räknesätten addition, subtraktion, multiplikation och division. Man behöver då bara förprogrammera Woodruff-transformationens utseende (bland annat de partiella derivatorna) för operationerna addition, subtraktion, multiplikation och division mellan två totaler. Det är så CLAN är uppbyggt som framgår av avsnitt 3 där macrona *%ADD*, *%SUB*, *%MULT* och *%DIV* presenteras.

Vi avslutar detta avsnitt med att på ett mer matematiskt sätt rättfärdiga tekniken med stegvisa Woodruff-transformationer.

Funktionen f i formel (6.1) antages vara rationell. Den kan då (utom i vissa triviala fall) delas upp på följande sätt för lämpligt valda f_0 , f_1 och f_2 ,

$$\hat{Y} = f_0(\hat{Y}_1, \hat{Y}_2)$$

där

$$\hat{Y}_1 = f_1(\hat{X}_1, \dots, \hat{X}_q, \dots, \hat{X}_Q) \text{ och } \hat{Y}_2 = f_2(\hat{X}_1, \dots, \hat{X}_q, \dots, \hat{X}_Q) \quad (6.12)$$

Transformationen för Y_1 och Y_2 blir,

$$z_{1i} = \sum_{q=1}^Q \frac{\partial Y_1}{\partial X_q} \cdot x_{qi} \text{ och } z_{2i} = \sum_{q=1}^Q \frac{\partial Y_2}{\partial X_q} \cdot x_{qi} \quad (6.13)$$

Transformationen för hela funktionen Y blir,

$$z_i = \sum_{q=1}^Q \frac{\partial Y}{\partial X_q} \cdot x_{qi} \quad (6.14)$$

Men enligt kedjeregeln gäller,

$$\frac{\partial Y}{\partial X_q} = \frac{\partial Y}{\partial Y_1} \cdot \frac{\partial Y_1}{\partial X_q} + \frac{\partial Y}{\partial Y_2} \cdot \frac{\partial Y_2}{\partial X_q} \quad (6.15)$$

Insättning av (6.13) och (6.15) i (6.14) ger då,

$$z_i = \frac{\partial Y}{\partial Y_1} \cdot z_{1i} + \frac{\partial Y}{\partial Y_2} \cdot z_{2i} \quad (6.16)$$

Man kan alltså åstadkomma transformationen (6.14) stegvis.

6.4. Variansformler under strategi 1-4

Genom användning av Woodruff-transformationen överförs alla variansberäkningar i CLAN på problemet att skatta varianser för totaler. Vi behöver alltså bara sådana formler.

Strategi 1

Låt N_h vara antal objekt i bruttopopulationen (eg urvalsramen) som tillhör stratum h , ($h=1, 2, \dots, H$) och m_h vara antalet svarande i stratum h *inklusive övertäckningen* (eventuella övertäckningsobjekt erhåller variabelvärdena $=0$).

Totalen Y skattas med,

$$\hat{Y} = \sum_{h=1}^H \frac{N_h}{m_h} \sum_i y_{hi} \quad (6.17)$$

och variansen för $\hat{Y}$ skattas med,

$$\hat{V}(\hat{Y}) = \sum_{h=1}^H \frac{N_h^2}{m_h} \left(1 - \frac{m_h}{N_h}\right) \frac{1}{m_h - 1} \left[\sum_{i=1}^{m_h} y_{hi}^2 - \frac{\left(\sum_{i=1}^{m_h} y_{hi}\right)^2}{m_h} \right] \quad (6.18)$$

Strategi 2

Stratum h delas in i L_h svarshomogenitetsgrupper. Objekten i en given grupp antages svara med samma sannolikhet, oberoende av varandra.

Populationsstorlekarna i grupperna antages ej kända.

I stratum h känner vi,

N_h antal objekt i bruttopopulationen (inklusive övertäckning),
 n_h antal utvalda objekt.

I svarshomogenitetsgrupp hg ($g=1, 2, \dots, L_h$), grupp g i stratum h , känner vi,

n_{hg} antal objekt i urvalet i stratum h som tillhör grupp g ,
 m_{hg} antal svarande i grupp hg .

Totalen Y skattas med,

$$\hat{Y} = \sum_{h=1}^H \frac{N_h}{n_h} \sum_{g=1}^{L_h} \frac{n_{hg}}{m_{hg}} \sum_{i=1}^{m_{hg}} y_{hgi} \quad (6.19)$$

Vi kan också skriva (6.19) som

$$\hat{Y} = \sum_{h=1}^H \sum_{g=1}^{L_h} \frac{\hat{N}_{hg}}{m_{hg}} \sum_{i=1}^{m_{hg}} y_{hgi} = \sum_{h=1}^H \hat{Y}_{hg} \quad (6.20)$$

där $\hat{N}_{hg} = \frac{N_h}{n_h} n_{hg}$ är en väntevärdesriktig skattning av det okända N_{hg} .

Variansen för $\hat{Y}$ skattas med, (se Särndal et al (1992) sid 582),

$$\begin{aligned} \hat{V}(\hat{Y}) = & \sum_{h=1}^H N_h(N_h-1) \left[\sum_{g=1}^{L_h} \frac{n_{hg}}{n_h} \left(\frac{n_{hg}-1}{n_h-1} - \frac{m_{hg}-1}{N_h-1} \right) \frac{1}{m_{hg}} s_{yhg}^2 \right] + \\ & + \sum_{h=1}^H N_h \frac{N_h - n_h}{n_h - 1} \sum_{g=1}^{L_h} \frac{n_{hg}}{n_h} (\bar{y}_{hg} - \bar{\bar{y}}_h)^2 \end{aligned} \quad (6.21)$$

$$\text{där } \bar{y}_{hg} = \frac{1}{m_{hg}} \sum_{i=1}^{m_{hg}} y_{hgi}, \quad \bar{\bar{y}}_h = \sum_{g=1}^{L_h} \frac{n_{hg}}{n_h} \bar{y}_{hg} \text{ och } s_{yhg}^2 = \frac{1}{m_{hg}-1} \left[\sum_{i=1}^{m_{hg}} y_{hgi}^2 - \frac{\left(\sum_{i=1}^{m_{hg}} y_{hgi} \right)^2}{m_{hg}} \right].$$

Strategi 3 och 4

Varje objekt utgör här ett kluster av ett eller flera element. Bland dessa element görs totalundersökning.

På samma sätt som tidigare antages urvalsramen bestå av N objekt/kluster som indelats i H strata, där stratum h innehåller N_h objekt/kluster.

I varje stratum dras OSU-UÅ n_h objekt/kluster av vilka vi får svar från m_h stycken.

Med övertäckning av objekt/kluster avses de objekt/kluster som inte innehåller några element som tillhör undersökningspopulationen.

Samma antagande om objektsbortfallet görs som tidigare.

Ett exempel på en sådan här situation är när urvalsobjekten/klustren är företag och elementen utgörs av arbetsställen till respektive företag. Arbetsställen inom samma företag kan tillhöra olika redovisningsgrupper.

Strategi 3 motsvarar strategi 1, dvs ingen indelning av objekten/klustren i svarshomogenitetsgrupper, alternativt indelning i grupper med kända gruppstorlekar.

Strategi 4 motsvarar strategi 2 med indelning av objekten/klustren i svarshomogenitetsgrupper med okända gruppstorlekar.

Formlerna (6.17)–(6.21) gäller som tidigare, skillnaden är definitionen av y -variabeln.

I strategi 3 definieras y_{hi} ,

$$y_{hi} = \sum_{k=1}^{K_{hi}} y_{hik}, \text{ där } K_{hi} \text{ är antalet element i kluster } hi.$$

I strategi 4 definieras y_{hgi} ,

$$y_{hgi} = \sum_{k=1}^{K_{hgi}} y_{hgik}, \text{ där } K_{hgi} \text{ är antalet element i kluster } hgi.$$

Enkelt uttryckt innebär strategi 3 och 4 att variabelvärdena för elementen inom ett kluster först summeras till klusternivå. Summorna utnyttjas sedan som variabler för punkt och medelfelskattningar på samma sätt som i strategi 1 och 2.

7. Referenser

Andersson, C., Augustsson, L. och Bengtsson, T. (1986). SMED83. PGMDOK 1986–10–07, U/ADB, SCB.

Nordberg, L. (1989). Om vidareutveckling av programvara för variansberäkningar vid SCB. PM 89–04–24, U/STM, SCB.

Särndal, C. E., Swensson, B. and Wretman, J. (1992). *Model assisted survey sampling*. New York: Springer-Verlag.

Woodruff, R. S. (1971). A simple method for approximating the variance of a complicated estimate. *JASA* 66, 411–414.

R & D Reports är en för U/ADB och U/STM gemensam publikationsserie, som fr o m 1988-01-01 ersätter de tidigare "gula" och "gröna" serierna. I serien ingår även **Abstracts** (sammanfattning av metodrapporter från SCB).

R & D Reports Statistics Sweden are published by the Department of Research & Development within Statistics Sweden. Reports dealing with statistical methods have green (grön) covers. Reports dealing with EDP methods have yellow (gul) covers. In addition, abstracts are published three times a year (light brown/beige covers).

Reports published during 1992:

- | | |
|-------------------|--|
| 1992:1
(grön) | Industrins konkurrenskraft och produktivitet i fokus - en utvärdering av statistiken (Margareta Ringquist) |
| 1992:2
(grön) | Automated Coding of Survey Responses: An International Review (Lars Lyberg and Pat Dean) |
| 1992:3
(grön) | TABELLER ,... TABELLER ,... TABELLER ,... - Variation och Förnyelse (Per Nilsson) |
| 1992:4
(grön) | Basurval vid SCB? Studier av reskostnadseffekter vid övergång till basurval (Elisabet Berglund) |
| 1992:5
(beige) | Abstracts I - sammanfattning av metodrapporter från SCB |
| 1992:6
(grön) | Utvärdering av framskrivningsförfarande för UVAV-statistik (Kerstin Forssén & Bengt Rosén) |
| 1992:7
(grön) | Cross-Classified Sampling for the Consumer Price Index (Esbjörn Ohlsson) |
| 1992:8
(grön) | Bortfallsbarometern nr 7 (Mats Bergdahl, Pär Brundell, Anders Lindberg, Håkan Lindén, Peter Lundquist, Monica Rennermalm) |
| 1992:9
(beige) | Abstracts II - sammanfattning av metodrapporter från SCB |
| 1992:10
(gul) | Organizing the Metainformation Systems of a Statistical Office (Bo Sundgren) |

Kvarvarande beige och gröna exemplar av ovanstående promemorior kan rekvireras från Inga-Lill Pettersson, U/LEDN, SCB, 115 81 STOCKHOLM, eller per telefon 08-783 49 56.

Kvarvarande gula exemplar kan rekvireras från Ingvar Andersson, U/LEDN, SCB, 115 81 STOCKHOLM, eller per telefon 08-783 41 47.