Bilaga 4 - Resultatrapport IT-verktyg för röjandekontroll etapp 3

# Systembeskrivning SAS2ARGUS

# Ett SAS-makro för att exekvera τ-Argus

Detta är en beskrivning av ett SAS-makro – SAS2Argus – som är framtaget för att underlätta riskbedömning av tabeller och undertrycka celler som inte går att publicera i tabeller som skall vara integritetsskyddade.

Innehåll

[Systembeskrivning SAS2ARGUS 1](#_Toc296082921)

[1 Ett SAS-makro för att exekvera τ-Argus 1](#_Toc296082922)

[1.1 Rekommenderad ”pedagogik” 3](#_Toc296082923)

[1.2 Val av metod 3](#_Toc296082924)

[1.3 Syftet med SAS-makrot 4](#_Toc296082925)

[1.3.1 Begrepp 5](#_Toc296082926)

[1.4 Huvudmakrot – SAS2Argus 5](#_Toc296082927)

[2 SAS2Argus – Beskrivning 6](#_Toc296082928)

[2.1 Parametrar 6](#_Toc296082929)

[2.2 Hierarkier 10](#_Toc296082930)

[2.3 Riskbedömning – SAFETYRULE 11](#_Toc296082931)

[2.4 Sekundärundertryckning – SUPPRESS 14](#_Toc296082932)

[2.4.1 A priori fil 15](#_Toc296082933)

[3 SAS2Argus - Användning 16](#_Toc296082934)

[3.1 SAS2Argus – Exempel på syntax 18](#_Toc296082935)

[3.1.1 Aggregerad tabell 18](#_Toc296082936)

[3.1.2 Arbetsgång 20](#_Toc296082937)

[3.1.3 Mikrodatatabell – magnitudtabell 25](#_Toc296082938)

[3.1.4 Mikrodatatabell – magnitudtabell 27](#_Toc296082939)

[4 SAS2Argus - Struktur 28](#_Toc296082940)

[4.1 Kontext 29](#_Toc296082941)

Denna beskrivning är en kortfattad sammanfattning av makrot som gör det möjligt, att produktionssätta röjande i en processkedja med hjälp av SAS. Ämnet är dock komplext och data/tabeller måste oftast beredas på ett särskilt sätt för att det ska vara möjligt att integritetsskydda tabeller säkert och rationellt. Det går inte att bara kan ”kasta” en producerad tabell på en röjandefunktion och tro att man får tillbaka den behandlad ur ett röjande­perspektiv färdig för publicering. Det är mycket som först måste ”berättas”.

## Rekommenderad ”pedagogik”

För att förstå syftet och användningen av SAS-makrot SAS2Argus, så under­lättar det väsentligt att ta det i bruk om man vet hur τ-Argus fungerar och används. Begrepp definierade och använda i τ-Argus används också inom SAS-makrot i så stor utsträckning som möjligt för att underlätta förståelsen för hur det ”hänger ihop”. Denna beskrivning av ”bryggan” mellan SAS och τ-Argus – hur ambitiös den än vill vara – haltar utan kunskap om principerna och metoderna i τ-Argus.

En rekommendation för en novis är att ”börja i τ-Argus” och bekanta sig med programmet och ta del av τ-Argus-manualen. Det underlättar väsentligt för­ståelsen för hela konceptet.

Vill man dessutom – eller har behov av att – sätta sig in i implementerade metoder, för att kunna välja ”rätt”, så rekommenderas även metodhandboken där dessa metoder beskrivs.

## Val av metod

Det finns en mängd metoder att välja mellan inom statistisk röjandekontroll som kan användas för att kunna framställa integritetsskyddade tabeller. Valet av metod beror på en mängd faktorer som hur data används, hur metoden går att implementera ur ett rent praktiskt perspektiv och vilket skydd den slutligen ger av tabellen vid publicering.

Metoder kan indelas i tre olika kategorier:

1. Pre-tabular Metoder som justerar mikrodata innan tabell-

framställningen görs

1. Table redesign Metoder som modifierar utformning av tabellen

genom att definiera om nivåer och detaljrikedom

1. Post-tabular Metoder som modifierar värden i en redan

framställd tabell

Att ”angripa” en redan framställd tabell – genom att applicera olika under­tryckningsmetoder på tabellens värden – hanteras inom den tredje kategorin, det vill säga post-tabulära metoder. Det är detta som kopp­lingen mellan SAS och τ-Argus erbjuder som koncept.

Det finns dock situationer när det är betydligt enklare att lösa problemet genom att överväga andra lösningar som ligger tidigare i processkedjan. Till exempel definiera om tabellen med större geografiska områden, kollaps av grupper och nivåer där antalet bidragsgivare är få. Fördelarna är flera. Metoderna är enkla att implementera och ger dessutom begripliga tabeller som är summerbara och det är lätt att förklara för användaren. Ibland är det dock inte möjligt om definitionen av tabellernas redovisning är fastställd eller att även denna ansats ger för mycket informationsförlust.

Pre-tabulära metoder som justerar värden på mikrodata, innan tabellfram­ställningen, ger summerbara tabeller men kan vara svårare, att för använ­daren beskriva använda datajusteringsmetoder.

Val av metod för skyddande av uppgiftslämnare är ändå något som måste övervägas, då det så påtagligt kan komma att påverka både arbetsinsatsen, komplexiteten i lösningen och den slutliga effekten.

## Syftet med SAS-makrot

Applikationsgränssnittet i τ-Argus arbetar med textfiler som måste skapas. Primära textfiler som behöver framställas för att ”sätta τ-Argus i arbete” är:

1. Datafil med antingen mikrodata eller aggregerade data [CSV]
2. Metadatafil som beskriver den första filens data [RDA]
3. Kommandofil som beskriver vilka regler för riskbedömning respektive undertryckning som skall användas och vilken typ(er) av resultatdatafil(er) [ARB] som skall produceras.

Syftet med makrot är att underlätta för användaren att komma åt funktionali­teten i τ-Argus genom att automatisera framställandet av dessa nödvändiga textfiler. Med användning av den metadatainformation som finns att tillgå i ett SAS-dataset, eller en SAS-vy mot annat data, (till exempel SQL-tabeller), kan mycket metadatainformation härledas som datatyp, antal decimaler med mera. Sen behöver användaren endast komplettera med information om variablers roller, riskbedömningsmetod och skyddsmetod.

Utöver de tre nämnda primära textfilerna kan ytterligare textfiler behöva skapas för att beskriva till exempel hierarkier i data, (*Hiearchy file*) [HRC], i fall detta är tillämpligt. Det går att med textfiler definiera etiketter på värden, (*Codelist file*) [CDL]. Det går också i form av textfiler att omkoda data. Då det är uppenbart enklare att inom SAS både etikettera och omkoda, så är denna möjlighet inte implementerad inom makrot. Det finns ytterligare en textfil definierad där man kan ange egenskaper på celler innan sekundär­undertryckningen utförs, en så kallad A Priori fil (*The apriori file*). Alla dessa beskrivs i τ-Argus-manualen.

### Begrepp

För att underlätta mappningen mellan begrepp som används inom τ-Argus och begrepp definierade i SAS-makrot SAS2Argus, så har samma begrepp återanvänds inom makrot, även om en del termer är ovanliga för oss, som till exem­pel explanatory[[1]](#footnote-1). På samma sätt används samtliga de suffix på filtyper som definieras och refereras i τ-Argus, för att underlätta läsning av τ-Argus-manualen och förstå mappningen mot SAS-makrot.

Makrot består av ett huvudmakro – SAS2Argus – och en uppsättning utility-makron som anropas inom huvudmakrot. Den ”normala” användaren ska egentligen inte behöva ”bry sig” om dessa submakron, men finns listade och beskrivna i slutet av detta dokument, för att i någon mån beskriva de ingående delarna i huvudmakrots funktion.

## Huvudmakrot – SAS2Argus

Användaren behöver bara förstå användningen av det övergripande SAS-makro, SAS2Argus, som alltså vid exekvering anropar en sekvens av andra makron som bland annat kont­rollerar syntaxen, genererar samtliga nödvän­diga inputfiler och startar en batch­körning av τ-Argus. Makrots funktion går således i stort sett ut på att från SAS etablera de text­filer som τ-Argus kräver för att köras med hjälp av det applikationsgränssnitt (API) som finns till­gängligt i τ-Argus, och därefter eventuellt även importera resultatet till SAS efter exekve­ring. Med förståelse för makrot och hur det parametersätts och hur τ-Argus funge­rar, är det relativt enkelt att utföra röjande med hjälp av makrot.

Alter­nativet är annars att exekvera τ-Argus genom det grafiska gränssnittet (GUI) – vilket kan vara en rekommendation till ”nybörjaren”, då det kan ge en ökad förstå­else för problem omkring röjande i allmänhet och de funktio­ner som finns att tillgå inom τ-Argus i synnerhet. Man kommer dock inte ifrån att först fabricera de textfiler i det format och med det innehåll som τ-Argus förväntar sig.

# SAS2Argus – Beskrivning

Här följer en beskrivning av makrot **SAS2Argus**. Hur man sätter upp SAS-sessionen för att komma åt funktionaliteten i SAS-makrot för att ”nå” funk­tionaliteten i τ-Argus. Vi börjar med att först beskriva vilka parametrar som makrot hanterar, då det i sig beskriver mycket av möjligheterna inom konceptet.

## Parametrar

Makrot är ett så kallat ”Named style macro” – det vill säga det har en upp­sättning namngivna parametrar som tilldelas värden som argument.

Notera att den följande beskrivningen av parametrar är en bruttouppsättning av makrots möjliga parametrar och att det i praktiken endast är ett fåtal som behöver specificeras. De flesta beskriver roller för variabler, som då också måste existera eller framställas manuellt i förväg för att senare kunna refere­ras. Några är dessa roller är ovanliga och används sällan. Ett antal parametrar åsätts defaultvärden, om värde inte anges explicit. Obligatoriska parametrar är mar­kerade med **färg**. Parameterlistan är uppdelad i följande kategorier:

* Generella parametrar (system parametrar)
* Parametrar som definierar Input data till τ-Argus
* Parametrar för riskbedömning och sekundärundertryckning
* Variabler och deras roller
  + Generella för både mikrodata och aggregerade data
  + Specifika för aggregerade data
  + Specifika för mikrodata
* Val av output från τ-Argus

Som det är ett ”Named style macro” så åtskiljs ***parametrar*** med komma­tecken, vilket gör att kommatecken ***inte*** skall användas för att åtskilja till exempel uppräknade variabelnamn ***inom*** en parameter.

För att göra parametrarnas kontext mer begriplig, så ges här kort ett exempel på hur ett anrop av makrot kan se ut, utan kommentarer i övrigt:

%***sas2argus***(InTable = APPdat.Freqdata,

Jobname = Tab\_1,

Explanatory = sex fam age ink,

Frequency = Resp,

SafetyRule = FREQ(**5**,**30**),

Out = inter(**1**),

RunArgus = **1**,

SAS = **2**,

Debug = **0**)

Notera att i nedanstående parameterbeskrivningen ges även en del anvisningar.

|  |  |
| --- | --- |
| **Parameter** | **Beskrivning** |
| ***Generella parametrar (system parametrar)*** | |
| **JOBNAME** | Ett namn som används som prefix for alla textfiler som skapas för τ-Argus och av τ-Argus inom ett ”job”/exekvering. Default, om inte angivet, så används **SAS2ARGUS**. Detta gör det enklare att i katalogen ”se” vilka filer som ”hör ihop” i en identifierbar kontext. |
| **RUNARGUS** | En option som möjliggör att styra om:   1. Endast textfiler skapas av makrot. τ-Argus exekveras inte. 2. Textfiler skapas och τ-Argus exekveras (**default**) 3. Skapa inga textfiler – exekvera endast τ-Argus på redan skapade textfiler   Detta gör det i praktiken möjligt att framställa filerna först, eventuellt editera textfilerna och slutligen exekvera de editerade textfilerna. |
| **DEBUG** | En options om ger möjlighet att få ut mer information till SAS-loggen:   1. Ingen extra information till SAS-loggen 2. Information skrivs till SAS-loggen, inklusive loggen från τ-Argus som också inkluderas i SAS-loggen (**default**)   Underlättar debugging och dokumentation då all tillgänglig information från körningen då återfinns i SAS-loggen. |
| **HELP** | Beskriver makrot och dess parametrar i SAS-loggen:   1. Ingen info i loggen (**default**) 2. Makrot beskrivs i loggen. Ingen annat utförs   Makrot är visserligen dokumenterat i skript-koden, men detta är ett enkelt sätt att få tillgång till en kortfattad beskrivning |
| **SAS**  Ny val fr.o.m. version 4.0 | En option som styr importen från τ-Argus till SAS:   1. Ingen import från τ-Argus till SAS (**default**) 2. Importerar resultatrapporten i HTML-format från τ-Argus och skriver ut den i webbläsaren. 3. Importerar output från τ-Argus till SAS WORK. 4. Importerar resultatrapporten och importerar output till SAS work. |
| ***Parametrar som definierar Input data till τ-Argus (En av dessa kan väljas, antingen INDATA eller INTABLE)*** | |
| **InData** | Specificerar namnet på SAS-dataset för mikro data. Notera att detta även kan vara en SQL-tabell. Alla datakällor som SAS stödjer med åtkomstmetoder kan användas. **Data** **måste anges**, antingen **InData** eller **InTable**. |
| **InTable** | Specificerar namnet på SAS-dataset för redan aggregerade data. Notera att detta även kan vara en SQL-tabell. Alla datakällor som SAS stödjer med åtkomst­metoder kan användas. Dock måste data ofta ”beredas” på något sätt och aggregerade data måste minst ha information om frekvens i respektive cell för att vara användbart som indata till riskbedömning och undertryckning. |
| ***Riskbedömning och sekundärundertryckning*** | |
| **SafetyRule** | Anger vilken metod för riskbedömning som skall användas. Detta/dessa argument kontrolleras inte i ”förebyggande” syfte av makrot. Studera τ-Argus-manualen för giltiga argument. Måste anges, om inte undantagsfallet gäller att cellernas status (variabelnamnet: Status) finns att tillgå och riskbedömningen således redan är gjord.  Se avsnitt [2.3 Riskbedömning - SAFETYRULE](#_Riskbedömning_–_SAFETYRULE) där denna parameter beskrivs. |
| **Suppress** | Anger vilken metod för undertryckning som skall användas. Detta/dessa argument kontrolleras inte i ”förebyggande” syfte av makrot. Studera τ-Argus-manualen för giltiga argument. Om detta argument utelämnas så innebär det att endast en riskbedömning görs.  Se avsnitt [2.4 Sekundärundertryckning - SUPRESS](#_Sekundärundertryckning_–_SUPPRESS_1) där denna parameter beskrivs. |

Informationen i SafetyRule och Suppress ”hamnar” i kommandofilen [ARB] för att berätta vad som skall göras. Resten av informationen från parametrarna nedan, används för att skapa datafilen [CSV], med dess netto­innehåll av nödvändiga/definierade roller/variabler, och metadatabeskriv­ningen av densamma [RDA].

|  |  |
| --- | --- |
| ***Variabler och deras roller – Generella för både mikrodata och aggregerade data*** | |
| **Explanatory** | Här anges namnet/namnen på de så kallade förklarande variablerna eller dimensions­variablerna som ”spänner upp tabellen”. **Måste anges**. Tillsammans med argumentet har två möjligheter implementerats. Om variabeln är hierarkisk så kan man i en efterföljande parentes tillfoga en beskrivning på vilket sätt den är hierarkisk eller i vilken textfil denna beskrivning kan återfinnas. Notera att variabelnamnen anges med blanksteg som avgränsare, om flera.  Se avsnitt [2.2 Hierarkier](#_Hierarkier) där detta beskrivs. |
| **Response** | Här anges namnet på responsvariabeln. **Måste anges**. |
| **Shadow** | Namnet på eventuellt skuggvariabel. Företagets omsättning kan vara en sådan ”hjälpvaria­bel”. Om den inte anges så använder τ-Argus Response-variabeln. |
| **Cost** | Namnet på eventuellt kostnadsvariabel. Vid sekundär undertryckning är det mindre sannolikt att en cell med hög kostnad blir undertryckt jämfört med en cell med låg kostnad. Om den inte anges så använder τ-Argus Response-variabeln. |
| **Lambda** | Transformationsparameter som används i en ”simplified Box Cox function” som exponent för kostnaden (Cost). **Default=1**. |

|  |  |
| --- | --- |
| ***Variabler och deras roller – Specifika för aggregerade data*** | |
| **Frequency** | Namnet på variabeln som beskriver frekvens. **Måste anges** för aggregerade data. |
| **LowerLevel** | Namnet på variabeln som anger lägsta nivå på ”skyddsintervall”. |
| **UpperLevel** | Namnet på variabeln som anger högsta nivå på ”skyddsintervall”. |
| **MaxScore** | Namnet på eventuella variabler som håller de enskilt högsta bidragsgivarna i respektive cell. Används i magnitudtabeller när dominansregeln tillämpas på i förväg aggregerade tabeller. Största bidragsgivare går att beräkna med PROC MEANS. Det finns ett utilitymakro som gör detta; **Calculate\_TopN.sas**. |
| **Status** | Namnet på eventuell variabel som anger status. Status (värde) kan då typiskt vara:  S = Safe  U = Unsafe  P = Protected  Det är rekommenderat att vara restriktiv att sätta cellstatus till protected. Om avsikten är att undvika att cellen blir föremål för en sekundär under­tryckning, så är det bättre att sätta en hög kostnad för cellen. |
| **TotCode** | En konstant som anger vilket värde som anger total i en aggregerad tabell. Default används tecken ”T”. |

För aggregerade data är det viktigt att inte glömma att ange parametern Frequency, då annars τ-Argus försöker aggregera redan aggregerade data för att fastställa frekvensen för antal bidrag i respektive cell. Status-variabeln ”tillverkas” av τ-Argus, när en riskbedöming görs, med regler angivna i parametern SafetyRule[[2]](#footnote-2), men går givetvis att skapa manuellt också.

|  |  |
| --- | --- |
| ***Variabler och deras roller – Specifika för mikrodata*** | |
| **Weight** | Namnet på variabeln som innehåller eventuell vikt. |
| **Holding** | Namnet på variabel som innehåller uppgift om koncern. Observationer som tillhör samma koncern skall grupperas tillsammans i indatafilen. |
| **Request** | Namnet på variabel som anger status på om respondenten har begärt skydd av uppgiften eller inte. Inversen till medgivande. |

Holding och Request har tillkommit efter användarkrav och används mest inom *Business Statistics*, respektive *Foreign Trade Statistics*.

|  |  |
| --- | --- |
| ***Val av output från τ-Argus*** | |
| **Out** | Möjliga val:  TABLE() => VarName delimiter(,) Primary(x) Secondary(-)  PIVOT(0) => VarName No Status  PIVOT(1) => VarName Status  CODE(0) => NoName delimiter(,) Primary(-) Secondary(x) No status  CODE(1) => NoName delimiter(,) Primary(del) Secondary(x) No status  CODE(2) => NoName delimiter(,) Primary(-) Status(1,5,11,14)  CODE(3) => NoName delimiter(,) Primary(del) Status(1,11)  SBS() => NoName delimiter(,) Exp,0,Exp,0.. zero(deleted) Status(V,D,A)  INTER(0) => NoName delimiter(;) Status only(S,M,U)  INTER(1) => NoName delimiter(;) Status(S,M,U)  Om parameter SAS=1 så importeras PIVOT and INTER (om valda). Enklast är att prova sig fram för att förstå alla olika alternativ för output. Den mest informativa och användbara får Intermediate (INTER) anses vara.  **Kommentar:** *VarName* innebär att variabelnamn återfinns. *NoName* inga varia­belnamn återfinns i textfilen. Tecken som används som avgränsare anges inom parentesen efter *Delimiter*. Tecken som ersätter primärt undertryckta värden anges i parentesen efter *Primary*. Tecken som ersätter sekundärt undertryckta värden anges inom parentesen efter *Secondary*. *Status/NoStatus* anger om cellstatus redovisas i outputen eller inte. |

Det är lätt att förstå att förklaringen till mängden output-format har en histo­risk förklaring, då det inom är ”τ-Argus-sfären” finns många intressenter som haft olika önskemål. SBS är till exempel ett specialtillämpat format för *Busi­ness Statistics*. De flesta formaten går inte, eller är olämpliga, att importera till SAS som en tabell, utan passar bäst att presentera i tabellform i till exem­pel Excel. Det mest informativa och användbara formatet är Inter­mediate, då det går dels att jobba interaktivt med, det vill säga redigera i SAS och skicka tillbaka till τ-Argus en andra gång, dels lätt att tabulera i SAS.

## Hierarkier

Hur man beskriver hierarkier kan behöva förklaras. Om man har att hantera hierarkiska data så kan dessa vara av två typer:

* Nivåer i ett värdeförråd
* Hierarkier som består av sammanslagningar av olika delar

Den första kan exemplifieras av *Län*, *Kommun*, *Församling* som i en sex­siffrig kod kan sägas beskriva nivå 1 i de två första positionerna, nivå 2 i de två nästkommande positionerna och slutligen nivå 3 i de sista 2.

Den andra kan exemplifieras med att ange *till exempel vilka län som utgör en landsregion* i listform. Hur en sådan lista kan se ut beskrivs i τ-Argus-manualen, och den textfilen (Hierachy file) [HRC], måste skapas manuellt. Dock kan filen sedan anges som ett tillägg till parametern Explanatory.

I makrot SAS2Argus så kan man beskriva den första typen av hierarki på följande vis:

Alt 1. Region(2 2 2)

…om det som i exemplet är fråga om *län*, *kommun*, *församling*. Begreppet inom τ-Argus för detta är <HIERLEVELS>.

Det andra alternativet som finns att tillgå är att ange ett filnamn på följande sätt:

Alt 2. Region(Region.hrc @)

…om hierarkin är beskriven i en fil med namnet Region.hrc. Det andra argumentet ’@’, (som är en option), anger då vilket tecken som skall använ­das som så kallad ”lead string”. Begreppet inom τ-Argus för detta är <HIERCODELIST>, respektive <HIERLEADSTRING>. Se τ-Argus-manualen för att förstå bättre hur hierarkier hanteras, om detta skulle vara aktuellt.

Det är annars mycket man slipper skriva själv i metadatabeskrivningen av hierarkier, då vi kan härleda följande ur dessa två argumentsvarianter:

<HIERARCHICAL> - Namnet på variabeln som är hierarkisk

…och antingen alternativ 1:

<HIERLEVELS> - Grupperingen inom en sträng

…eller alternativ 2:

<HIERCODELIST> - Namnet på filen som definierar hierarki

<HIERLEADSTRING> - Specialtecken att tolka i filen

## Riskbedömning – SAFETYRULE

I sekretessammanhang så får det anses som ganska enkelt att definiera vilka celler/domäner som osäkra att publicera. Går det att formulera en regel för vilka celler/domäner som inte kan publiceras, så är dessa också enkla att identifiera med relativt enkla verktyg/teknik.

I τ-Argus specificeras riskbedömning i parametern SafetyRule. Samma begrepp används i SAS-makrot. Argument som ges här skall anges på det sätt som beskrivs i τ-Argus-manualen. Någon kontroll på angivna argument görs inte av SAS-makrot. Information om eventuella felaktigheter i argumen­tationen kan då komma att redovisas i τ-Argus-loggen. (Tips: Sätt parameter DEBUG=1, så infogas τ-Argus-loggen i SAS-loggen.)

Parametern SafetyRule kan ta många argument och definierar primär­under­tryckningen, eller identifiering av primärt osäkra celler. Notera att flera primärundertryckningsprinciper kan anges, dessa separeras då med ”**|**”. Följande regler kan anges: P, NK, ZERO, FREQ, REQ, WGT, MIS och beskrivs i tabellen nedan. Tilläggsargument anges typiskt inom en efterföljande parentes:

|  |  |
| --- | --- |
| **SAFETYRULE** | **Beskrivning** |
| P | Procentregel där tilläggsargument anges som P(p,n) där n är option och default satt till 1. P(20,3) innebär procentregel där p=20 och n=3. |
| NK | Dominansregel där tilläggsargumentet anges som NK(n,k) där n anger antal objekt som inte får stå för mer än k procent av bidraget i cellen/domänen. |
| ZERO | Säkerhetsmarginal för 0-celler där tilläggsargumentet anges som ZERO(ZeroSafetyRange) och avser storlek på marginalen. |
| FREQ | Frekvensregeln där tilläggsargumentet anges som FREQ(MinFreq,FrequencySafetyRange) där MinFreq anger minimun acceptabel frekvens och FrequencySafetyRange anger med vilken marginal det inte ska gå att härleda den undertryckta frekvensen. |
| REQ | ”Request rule” – anhållan om sekretess.[[3]](#footnote-3)  REQ(Percent1,Percent2,Safetymargin). Om till exempel en uppgifts­lämnare står för 70 % och i det fallet har begärt skydd. Detta kräver en extra variabel i tabellen som anger vilka objekt som begärt/inte begärt sekretess med värde 1 eller 0. Variabelns namn (för rollen) anges med [parameter Request](#Request). |

|  |  |
| --- | --- |
| **SAFETYRULE** | **Beskrivning** |
| MIS | Missing. Om MIS=0 (vilket är default) innebär att celler med kod för bortfall betraktas fortfarande som osäkra om någon SafetyRule kränks. Om MIS=1 så betraktas cellen alltid som säker, om minst en bidragsgivare har bortfall, då det får anses som att bidragsgivaren med bortfall inte går att identi­fiera. I SAS-makrot har 9:or definierats som bortfallsvärde[[4]](#footnote-4). |
| WGT | Vikt. Om WGT=0 (vilket är default) så används inga vikter vid aggregering av tabeller eller vid beräkning av SafetyRule. Varibelns namn för rollen som vikt anges med [parameter Weight](#Weight). |
| MAN | ”Manual safety margin” (default=20%). Denna manuellt satta säker­hets­marginal används endast när status tillhanda­hålls för varje cell eller när en A priori fil[[5]](#footnote-5) används för att sätta optionen att en viss cell är manuellt osäker (Manual Unsafe). |

Alla regler kan förekomma flera gånger, avdelade med ”|”. Exempel:

NK(3,70)|FREQ(3,30)|MIS(1)

Förklaring:

* NK(3,70) innebär dominansregel, det vill säga om 3 observationer bidrar med mer än 70 procent så skall cellen betraktas som osäker.
* eller FREQ(3,30) innebär att om frekvensen är mindre än 3 och en inkräktare (som känner till sitt eget bidrag i redovisningen) inte ska inte kunna avslöja ett annat värde med mindre än 30 procents marginal. Då ska cellen betraktas som osäker.
* eller MIS(1) innebär att om något bidrag till cellen är bortfall så kan cellen betraktas som säker.

Alla regler kan, som sagt, anges flera gånger. De första två P och NK antas då gälla för den individuella nivån och de efterföljande P och NK antas då gälla för koncernnivån (som hanteras/definieras av [parameter Holding](#Holding)).

Den första FREQ och REQ antas vara för den individuella nivån och de efterföljande för koncernnivån.

ZERO kan endast anges en gång för varje säkerhetsregel.

## Sekundärundertryckning – SUPPRESS

Här beskrivs argument som kan användas för parametern Suppress. Den har samma ”syntax” som SafetyRule, det vill säga tilläggsargument anges inom en efterföljande parentes. Notera dock att den första av dessa tilläggs­para­metrar anger tabellnummer (TabNo) och som SAS-makrot endast han­terar en tabell i taget, så skall denna tilläggsparameter alltid vara en 1:a.

Följande metoder kan användas för sekundärundertryckning:

|  |  |
| --- | --- |
| **SUPPRESS** | **Beskrivning** |
| GH | GH(TabNo,AprioriBoundPercentage,ModelSize)  GH-miter eller som den oftast kallas Hypercube-metoden. Tillsammans med NET de enda metoder som finns tillgängliga inom τ-Argus utan tillgång till kommersiell optimerare. Den sekundärundertrycker ”mekaniskt” utan avseende på optimering och rekommenderas inte som metod då den inte sällan orsakar stor informationsförlust i tabellen. |
| MOD | MOD(TabNo,MaxTimePerSubtable)  En partiell metod som bryter ner en hierarkisk tabell i flera icke hierarkiska tabeller, skyddar dessa och komponerar slutligen en helt skyddad hierarkisk tabell. Maxtiden går att begränsa med parameter som sätter maxtiden för optimeraren att arbeta med varje subtabell. Anges i minuter. |
| OPT | OPT(TabNo,MaxComputingTime)  En metod som skyddar en hierarkisk tabell utan att bryta ner den i mindre tabeller. Maxtiden går att begränsa med parameter som sätter maxtiden för optimeraren att arbeta. Anges i minuter. |
| NET | NET(TabNo)  Network solution att användas för stora 2-dimensionella tabeller med en hierarki. Kräver vissa omständigheter men ingen optimerare. |
| RND | RND(TabNo,RoundingBase,Steps,1,Time,Partitions,   StopRule)   * RoundingBase avser basen för avrundning * Steps avser antal step tillåtna (**default=0**) * Fjärde argumentet är konstant 1 (för framtida utökning) * Partitions (**default=0**). 1 innebär inversen * StopRule (**default=3**) * 1 = Rapid only * 2 = First feasible solution * 3 = Optimal solution |

### A priori fil

En A priori fil är en enkel text-fil som kan skapas i en editor. Informa­tionen i A priori filen används *efter* riskbedömning, men *före* sekundärundertryck­ning. Filen uppdaterar bland annat Status, Cost, LowerLevel och UpperLevel och kan även sätta en nytt värde på skyddsnivå för enskilda celler om annan kunskap för enskilda celler finns till hands. Det vill säga vi ges en möjlighet att tillföra känd kunskap till tabellen efter den första riskbedömingen. SAS-makrot har för närvarande ingen parameter, (vilket i och för sig vore enkelt att lägga till), för att hantera en A priori fil, men det går att editera kommandofilen [ARB] genom att lägga till argumentet:

<APRIORI> ”Filename”, TabNo, Separator.

Varje rad inleds med kommaseparerade koder som identifierar varje enskild cell och följs därefter, efter ytterligare ett komma, med följande koder och värden för att tillföra information om känd kunskap för särskilda celler:

|  |  |  |
| --- | --- | --- |
| Kod | Parameter | Förklaring |
| S | - | Status ändras till safe |
| U | - | Status ändras till unsafe |
| P | - | Status ändrat till protected |
| C | Nytt värde för kostnad (COST) | En låg kostnad ökar sannolikheten att cellen blir kandidat för sekundär undertryckning. En hög kostnad ökar sannolikheten för detsamma. |
| PL | Två nya värden för LowerLevel och UpperLevel åtskilda med komma. | Om intervallet för en viss cell i förväg kan antas vara mindre än det normala intervallet så kan det anges här för att få en ny ”protection level” |

Not: Möjligheterna att ändra statusen på en cell är förståss begränsat. Det innebär bland annat att en primärt unsafe cell inte bör ändras till protected. Inte heller kan en protected cell ändras till unsafe.

Kostnaden (Cost) måste alltid vara ett positivt värde.

Det är rekommenderat att vara restriktiv att sätta cellstatus till protected. Om avsikten är att undvika att cellen blir föremål för en sekundär under­tryckning, så är det bättre att sätta en hög kostnad för cellen. Om den ändå skulle bli sekundärundertryckt av algoritmen i τ-Argus, så finns det goda skäl till detta.

Se i övrigt τ-Argus-manualen för att få en beskrivning på filens format, möjliga argument och exempel.

# SAS2Argus - Användning

För att initiera makrot behövs bara sättas en ”pekare” för att berätta för SAS var ”resursen” finns.

/\*==========================================================================

Initial path location on disk

--------------------------------------------------------------------------\*/

%let PATH\_ini=C:\SAS2Argus;

/\*==========================================================================

Path to the generic part and version and the application part. Also give

the full path to tau\_Argus executable [PATH\_exe]

Note: If PATH\_tmp is not set => we route to SAS Work as the working area.

Preferable PATH\_tmp is set during test and trouble shooting to be

able to examine files.

--------------------------------------------------------------------------\*/

%let PATH\_sys=&Path\_ini.\1.1;

%let PATH\_app=&Path\_ini.\Demo;

%let PATH\_tmp=&Path\_ini.\Demo\APPtmp;

%let PATH\_exe=C:\Program\TauArgus\tauARGUS.exe;

Sen finns ett litet initieringsprogram som anger sökvägen till SAS-program. Det enda generella SAS-program som används är A01 Init Session.sas som i sin tur anger sökvägen till SAS-makron, enligt principen som i SAS anges med AUTOSOURCE. Det innebär att alla makron kompileras i den stund de efterfrågas.

/\*==========================================================================

Init the SAS session

--------------------------------------------------------------------------\*/

options notes nomprint nomlogic;

filename SASpgm "&PATH\_sys.\SASpgm";

%include SASpgm("A01 Init Session.sas");

Sen gör vi bara en kontroll att allt finns på plats med ett makroanrop:

/\*==========================================================================

Macro: System\_Parameters

---------------------------------------------------------------------------

Checks if:

- full PATH to tau-Argus exe is present so we can find it

- EXE file for tau-Argus is present so we can use it

- PATH\_tmp is set explicit. Otherwise we route to SAS WORK-path

and reports finally the used tau-ARGUS work path to the SAS log

--------------------------------------------------------------------------\*/

%***System\_parameters***;

/\*==========================================================================

Here ends the initialisation of the SAS session for use of the bridge

between SAS and tau-Argus

==========================================================================\*/

Efter initieringen av den generella delen följer den specifika delen som företrädesvis börjar med att definiera ett libname mot något datakälla/ databas så att vi kan komma åt data:

/\*==========================================================================

Here beneth follows the specific part - the production part of the concept.

==========================================================================\*/

/\*==========================================================================

Libname to data

--------------------------------------------------------------------------\*/

libname APPdat "&PATH\_app.\APPdat";

Notera att libname kan också definiera en OLEDB-koppling mot SQL-server.

## SAS2Argus – Exempel på syntax

Här följer några exempel på syntax där man anropar makrot .

### Aggregerad tabell

Här ett exempel på ett aggregerad tabell (InTable) som riskbedöms med frekvensregeln, FREQ(5,30), att det skall vara minst 4 observationer i varje redovisningscell med 30% säkerhetsmarginal.

/\*==========================================================================

EXAMPLE 1 - FREQUENCY

---------------------------------------------------------------------------

Comment: An aggregated table only primary suppressed

--------------------------------------------------------------------------\*/

%***sas2argus***(InTable = APPdat.TauFreq,

Jobname = Example1,

Explanatory = sex fam age ink,

Frequency = Resp,

SafetyRule = FREQ(**5**,**30**),

Out = table() pivot(**1**) code(**3**) inter(**1**),

RunArgus = **1**,

SAS = **2**,

Debug = **1**

)

Det är viktigt – för en aggregerad tabell – att alltid ange Frequency, som ses i syntaxen ovan, då det är väsentlig information för τ-Argus för att utifrån en redan aggregerad tabell kunna göra en riskbedömning.

Namnstandarden på filer utgår från Jobname – som används som prefix på samtliga filer ”som ingår i en exekvering”. Följande filer produceras i ovanstående anrop för τ-Argus:

Example1.csv - Datafilen tabb-separerad

Example1.rda - Metadatafilen

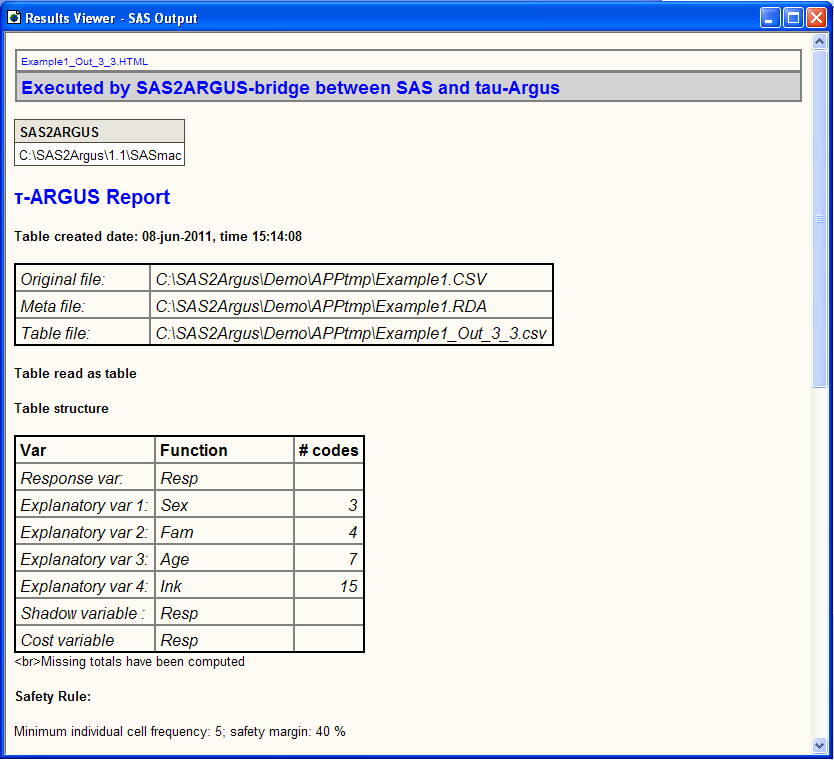
Example1.arb - Kommandofilen

Det finns också några hjälpmakron som underlättar att öppna och ”titta” på både input och output-filer. Open\_Editor öppnar vilken textfil som helst i SAS Enhanced Editor:

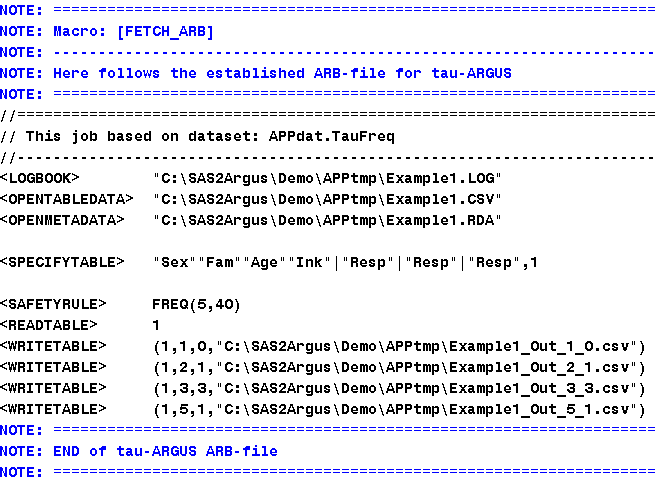
%***open\_editor***(Example1\_out\_5\_1.rda)

Här, i exemplet, öppnar vi filen Example1\_out\_5\_1.rda i SAS Enhanced Editor. Det är metadatabeskrivningen för den med namnet associerade datafilen Example1\_out\_5\_1.csv som producerats av τ-Argus. Ur namnet kan vi utläsa det är en outputfil av typ 5.1 från Example1. Det vill säga argumentet Inter(1) medför det 5:e outputformatet med bruttomängd av information och status (1)**.**

Med parameter SAS=2 så returneras/importeras resultatet till SAS-sessionen och den producerade datafilen återfinns i SAS Work. Resultatrapporten i form av en HTML-fil från τ-Argus, återfinns i SAS interna browser (lite beroende på inställningarna i SAS):



Med parameter Debug=1 så skrivs ARB-filen (kommandofilen) till SAS-loggen:



Likaså återfinns loggen från τ-Argus i SAS-loggen.

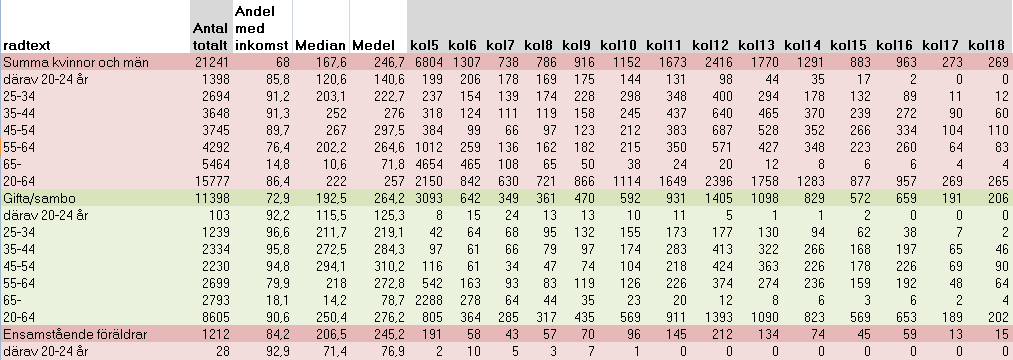
### Arbetsgång

Processen att integritetsskydda tabeller, och arbetsinsatsen, är beroende av hur dispositionen av tabellen, tabellerna, data är initialt. Det kan vara allt från mycket enkelt och redan tillrättalagt till riktigt komplicerat, att reorganisera data på ett sätt som passar för röjandekontroll. Lite av ”var kommer vi ifrån” och ”vart är vi på väg”.

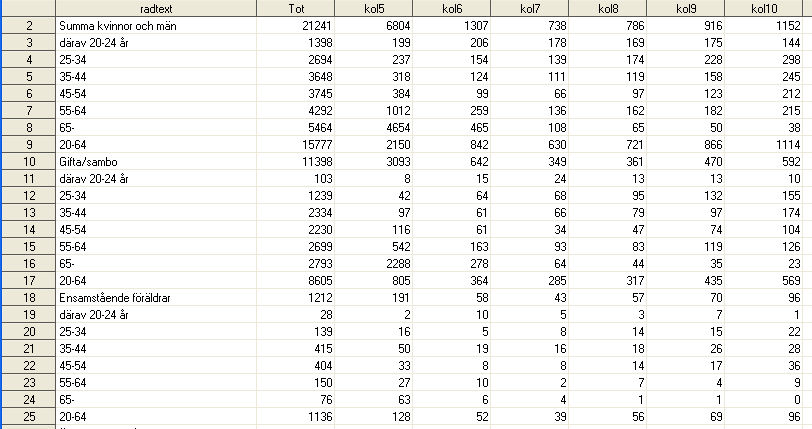
Variabler som behövs i olika ”roller” i röjandeprocessen måste finnas till hands och varje cell behöver beskrivas med egenskaper som ett unikt id, det vill säga dimensionsvariablernas värde för varje cell. Andra egenskaper kan utöver detta vara vikt, kostnad, skuggvariabel och så vidare.

Om tabelldata inte är orienterade i radledd så måste tabelldata struktureras om. Om vi utgår från en tabell från inkomststatistiken som innehåller dels flera olika statistiska mått, olika dimensionsvariabler (delvis överlappande dessutom) och antal i olika inkomstklasser.

Initialt kan en tabell se ut så här:

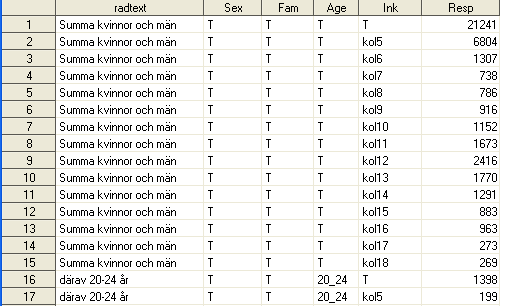


Om tabelldata inför röjandekontroll initialt ser ut så här, där vi för vårt exempel endast intresserar oss för de kolumner som är gråmarkerade som innehåller uppgifter om antal personer i olika inkomstklasser, och där dessa värden är orienterade i kolumnledd (från kol5 till kol18):



… så måste datafilen struktureras om – så att cellerna identifieras med dess egenskaper.

Vi får då hela tabellen i listform där varje cell utgör en rad:



Radtext är i ovanstående exempel visserligen överflödig infor­mation för τ-Argus, men i övrigt är nu tabellen användbar som indata till röjandeproces-sen och som redovisats tidigare, i det första exemplet av SAS2Argus-anrop.

Det finns mängder med tekniker att tillgå för att ”vrida” data och beror på val av verktyg, SQL (case, pivot…), SAS (transpose, group datastep…).

Vi har nu skapat dimensionsvariabler, (från den enda informationen tillgäng­lig i radtext) och har infört en konstant ”T” som betecknar totaler, det vill säga tabellens marginalsummor. De överlappande ålderklasserna (20-64) är *inte* hanterad, men kan utgöra ett exempel på att; om vi skulle hanterat även detta, så har vi här en hierarki, som i så fall också måste beskrivas för τ-Argus.

Om vi nu definierar ett anrop med makrot för att göra en riskbedömning av tabellen/tabellerna med kriteriet att det skall vara minst 5 observationer i varje cell i det radorienterad data vi nu berett. Vi definierar namnet på datasetet och följande roller, regler och systemparametrar:

%***sas2argus***(InTable = APPdat.TauFreq,

Jobname = Example1,

Explanatory = Sex Fam Age Ink,

Frequency = Resp,

SafetyRule = FREQ(**5**,**40**),

Out = inter(**1**),

RunArgus = **1**,

SAS = **2**,

Debug = **1**

)

Notera att vi för en aggregerad frekvenstabell inte behöver ange Response-variabel, utan Frequency=Resp räcker för att riskbedöma tabellen utfrån frekvens.

Systemparametern/optionen SAS=**2** innebär att vi även importerar importer­bara dataset till SAS. Intermeditate-tabellen fås som output av argumentet: Out=inter(**1**) , och som det är ett dataset möjligt att importera så återfinns det efter körningen i WORK och ser ut så här efter exekvering:

### 

Notera Statusvar som med värde S=Safe respektive U=Unsafe (när frekvensen är under 5).

Namnet på output-typen; Intermediate antyder att detta är det lämpligaste formatet att arbete med, då det går att antingen editera och skicka tillbaka till τ-Argus eller skicka vidare i processkedjan för att ställa upp presenterbara tabeller.

### Mikrodatatabell – magnitudtabell

Det förra exemplet var en aggregerad frekvenstabell. Exemplet här gäller en magnitudtabell, som vi låter SAS aggregera innan vi röjandekontrollerar den. Om vi önskar tillämpa den så kallade NK-regeln, N objekt får inte bidra med mer än K % av cellinnehållet, så måste vi identifiera de största bidrags­givarna först i varje cell. Detta går att göra med följande syntax i PROC MEANS:

/\*==========================================================================

Summarize and select four TopN values for the explanatory variables.

--------------------------------------------------------------------------\*/

**proc means** data=APPdat.TAUmicro Missing CompleteTypes NWay NoPrint;

class Age Omk;

var Resp;

output out=APPdat.TAUmicro\_agg

sum=Resp

idgroup(max(Resp) out[**4**] (Resp)=Resp\_TopN)/NoInherit;

**run**;

(…där de 4 största bidragsgivarna fås av den markerade syntaxen).

Eller, om man har svårt att komma ihåg syntaxen, använda makrot Calculate\_TopN (som återfinns bland utility-makron):

/\*--------------------------------------------------------------------------

Run the macro CALCULATE\_TopN.

--------------------------------------------------------------------------\*/

%***Calculate\_TopN***(Indata=APPdat.TAUmicro,

ClassVar=Age Omk,

Var=Resp,

TopN=**4**);

Genom att ”luta” oss mot den metadata som vi har att tillgå i SAS kan vi ange till exempel ett format för att berätta hur många decimaler som vi vill arbeta med. Då ”ser” makrot detta och vi får med det i specifikation när metadatafilen skapas:

/\*==========================================================================

Assign a SAS-format for "telling" SAS2ARGUS how many decimals we should

deal with when establishing the files for tau-ARGUS.

--------------------------------------------------------------------------\*/

**proc** **datasets** library=APPdat NoList;

modify TAUmicro\_agg;

format Resp: **8.3**;

**run**; **quit**;

Sen exekverar vi SAS2Argus-makrot:

/\*==========================================================================

EXAMPLE 2 - Magnitude

--------------------------------------------------------------------------\*/

options nonotes nosource nomprint nomlogic nosymbolgen;

%***sas2argus***(InTable = APPdat.TAUmicro\_agg,

Jobname = Example2,

Explanatory = Age Omk,

Response = Resp,

MaxScore = Resp\_TopN\_1 Resp\_TopN\_2 Resp\_TopN\_3 Resp\_TopN\_4,

SafetyRule = NK(**3**,**90**),

Suppress = GH(**1**,**40**,**0**),

Out = table() pivot(**1**) code(**3**) inter(**1**),

RunArgus = **1**,

SAS = **2**,

Debug = **1**

)

Här anger vi utöver Explanatory och Response också de med PROC MEANS framställda MaxScore-variablerna namn. SafetyRule är här NK(3,90), den så kallade NK-regeln som anger att 3 företag får inte bidra med mer än 90 % av innehållet i domänen/cellen. Sekundärundertryck­ningen, GH(1,40,0), , sker här med hyperkub-metoden (eller GH-miter som den också benämns). Där 1 står för ”tabell 1” (och att notera generellt att vi exekverar bara en tabell i taget via makrot). 40 står för 40 % och avser maxprecisionen som det eventuellt skulle kunna gå att beräkna undertryckt värde mot kvarvarande redovisade marginalsummor.

Hyperkub metoden tillsammans med NET-metoden, är de enda som finns att tillgå som sekundärundertryck­ningsmetoder, om man inte har tillgång till en optimerare. SCB har införskaffat en optimerare som ger fler alternativ. Hy­perkub metoden ”släcker” mycket av tabellens celler i sekundärunder­tryck­ningsfasen, då den arbetar ”mekaniskt”, och är inte en rekommen­derad som försthandsval av metod. Modular-metoden är då bättre och att förorda, då den försöker hitta den lösning som använder färst sekundära celler i sekun­därundertryckningen.

### Mikrodatatabell – magnitudtabell

Ett sista exempel visar en mikrodatatabell som innehåller befolkning och där län, kommun, församling, **Region(2 2 2)** utgör den ena hierarkin och ålder i klasser, **Age(&PATH\_app\APPdat\Age.hrc)**, utgör den andra hierarkin av åldersklasser. I det senare fallet finns den upprättad fil som beskriver ålders­klasshierarkin vars namn, tillsammans med sökväg anges som argument (inom parentes). Om HierLeadString inte anges (som här) antas ’@’. Sen återfinns en 3:e förklarande variabel kön (Sex). Response-variabeln (Count) är i detta fallet en kolumn med ettor.

/\*--------------------------------------------------------------------------

EXAMPLE 3 - HIERARCHY

--------------------------------------------------------------------------\*/

options nomprint nosource;

%***sas2argus***(InData = APPdat.Population,

Jobname = Example3,

Explanatory = Region(**2** **2** **2**) Age(&PATH\_app\APPdat\Age.hrc) Sex,

Response = Count,

SafetyRule = NK(**3**,**75**)|P(**25**,**100**,**1**)|FREQ(**5**,**30**),

Suppress = GH(1,30,0),

Out = table() inter(**1**),

RunArgus = **0**,

SAS = **1**,

Debug = **1**

)

Flera SafetyRules kan anges samtidigt som synes. Se τ-Argus-manualen för en detaljerad beskrivning av vilka argument som är möjliga.

RunArgus=0 innebär att endast de textfiler som τ-Argus behöver produ­ceras. τ-Argus exekveras inte. Dessa textfiler går att återanvända i det interaktiva gränssnittet.

SAS=1 har här ingen effekt då τ-Argus inte kommer att exekveras på grund av föregående argument. Skulle annars inneburit att resultatrapporten och loggen ”importerats” och presenterats inom SAS-sessionen.

# SAS2Argus - Struktur

SAS2Argus byggs upp av ett huvudmakro som använder sig av en uppsättning ”utility-makron”.

|  |  |
| --- | --- |
| **Makro** | **Beskrivning** |
| **Sas2Argus.sas** | Huvudmakrot som användaren parametersätter och exekverar |
| ***Makron som anropas av makrot SAS2Argus:*** | |
| SAS2Argus\_Help.sas | Skriver hjälp till loggen |
| Clean\_Parameters.sas | Städar parametervärden |
| Check\_Parameters.sas | Kontrollerar parametervärden |
| Variable\_Roles.sas | Fastställer variablers roller |
| Variable\_Properties.sas | Fastställer variablers egenskaper |
| Variable\_Meta.sas | Kontrollerar att angivna variabler existerar i angiven datatabell (SAS, SQL, Excel…) |
| Write\_Datafile.sas | Skapar textfil från tabelldata |
| Write\_Jobfile.sas | Skapar kommandofil |
| Fetch\_Arb.sas | Inkluderar kommandofilen i SAS-loggen |
| Remove\_File.sas | Tar bort eventuell logg-fil från tidigare körningar |
| Read\_Datafile.sas | Importerar textfil från τ-Argus till SAS |
| Present\_HTML.sas | Presenterar resultatrapporten i SAS interna browser |
| Argus2SAS.sas | Importerar outputen från τ-Argus till SAS genom att exekvera Read\_Datafile.sas och Present\_HTML.sas |
| Fetch\_Log.sas | Presenterar τ-Argus loggen i SAS-loggen |
| ***Makro som kontrollerar uppsättningen av sessionen:*** | |
| System\_Parameters.sas | Initialt makro som kontrollerar de initialt nödvändiga parametrar som behövs för att sätta upp funktionaliteten i SAS |
| ***Andra användbara hjälpmakron (utility-makron):*** | |
| Calculate\_TopN.sas | Ett utilitymakro som aggregerar (PROC MEANS) och som redovisar de största bidragsgivarna i respektive cell |
| Check\_Dataset.sas | Ett utilitymakro som kan kontrollera existensen av ett dataset, vy, tabell på SQL-server… |
| Check\_Outstring.sas | Ett utilitymakro som kontrollerar att rätt argument angetts för output från τ-Argus |
| Open\_Editor.sas | Ett utilitymakro som öppnar valfri textfil i SAS programeditor |
| Open\_Excel.sas | Ett utilitymakro som öppnar valfri CSV-fil i Excel |

## Kontext

För att beskriva kontexten är det enklast att – i kondenserad form – redovisa själva huvudmakrot SAS2Argus och hur det i sin tur exekverar ”under­liggande” utility-makron för den som vill förstå ur ett systemperspektiv.

**%macro SAS2Argus**(

/\*----------------------------------------------------------------------

*Need for help?*

-----------------------------------------------------------------------\*/

%if **&help.** %then %do;

**%SAS2Argus\_help**;

%end;

/\*----------------------------------------------------------------------

*"Clean" the parameters*

-----------------------------------------------------------------------\*/

%let Explanatory = **%Clean\_Parameters**(&Explanatory.,p);

…

…

/\*----------------------------------------------------------------------

*Check parameters for data and that the dataset exists*

-----------------------------------------------------------------------\*/

**%Check\_parameters**;

/\*----------------------------------------------------------------------

*If parameter RUNARGUS is set to 2 => Rerun τ-Argus with already created*

*files meaning that there is no need to establish any CSV or RDA or ARB-*

*files. This is the scenario when the user could changed the status on*

*certain domains (cells) from SAFE to UNSAFE or from UNSAFE to SAFE and*

*with a new request to τ-Argus to handle secondary cell suppression.*

-----------------------------------------------------------------------\*/

%if **&runargus.** ne 2 %then %do;

/\*-------------------------------------------------------------------

*Establish the Metadata for the ROLES OF VARIABLES at macro invocation.*

--------------------------------------------------------------------\*/

**%Variable\_Roles**

/\*--------------------------------------------------------------------

*Establish the Metadata for the ACTUAL VARIABLES in the dataset given at*

*macro invocation.*

--------------------------------------------------------------------\*/

**%Variable\_Properties**(&\_dataset.)

/\*-------------------------------------------------------------------

*Check conformity between designated variable roles and actual variables*

*in the dataset.*

--------------------------------------------------------------------\*/

**%Variable\_Meta**(&\_dataset.)

/\*-------------------------------------------------------------------

*Create the textfiles .RDA (meta) and .CSV (data ) according to the*

*information established so far and found in \_Variable\_Meta*

--------------------------------------------------------------------\*/

**%Write\_Datafile**(Dataset=%bquote(&\_dataset.),Datafile=&PATH\_tmp.\&jobname.)

/\*-------------------------------------------------------------------

*Create the Command file (BAT) for τ-Argus as a manifest of what we*

*want τ-Argus to do (suffix .ARB)*

--------------------------------------------------------------------\*/

**%Write\_Jobfile**(Jobfile=&PATH\_tmp.\&jobname..ARB)

%end;

/\*----------------------------------------------------------------------

*If debug=1 then include the ARB-file in the SAS log*

-----------------------------------------------------------------------\*/

%if **&debug.** %then %do;

**%Fetch\_Arb**(Arbfile=&PATH\_tmp.\&jobname..ARB)

%end;

/\*----------------------------------------------------------------------

*Execute τ-Argus*

-----------------------------------------------------------------------\*/

%if **&RunArgus.** %then %do;

/\*-------------------------------------------------------------------

*Remove the LOG for this "job"*

--------------------------------------------------------------------\*/

**%Remove\_File**

/\*-------------------------------------------------------------------

*Execute τ-Argus*

--------------------------------------------------------------------\*/

data \_null\_;

\_cmd = """&PATH\_exe"" ""&PATH\_tmp.\&jobname..ARB""";

**call system(\_cmd);**

run;

/\*-------------------------------------------------------------------

*If SAS=1 then "import" both TEXT files and HTML files produced by*

*τ-Argus to this SAS-session.*

---------------------------------------------------------------------

**%Argus2SAS** *executes:*

**%Read\_Datafile** *- A macro that reads delimited (CSV) text files*

*with use of supplied metadata file (RDA) and*

*establish SAS datasets.*

**%Present\_*HTML*** *- A macro that presents the HTML-file produced by*

*τ-Argus in the SAS internal browser.*

--------------------------------------------------------------------\*/

%if **&SAS.** %then %do;

**%Argus2SAS**

%end;

/\*-------------------------------------------------------------------

*If debug=1 then include the HTML-files from τ-Argus in the SAS*

*internal browser and the LOG-file from τ-Argus in the SAS log*

--------------------------------------------------------------------\*/

%if **&debug**. %then %do;

**%Fetch\_Log**

%end;

%end;

%mend SAS2Argus;

Samtliga ingående makron är väl kommenterade för att öka förståelsen och underlätta för framtida systemunderhåll.

1. ”Explanatory” används (och/eller tillsammans med ”spanning” i manualen) som begrepp för att ange/beskriva dimen­sionsvariabler och som internt i τ-Argus översätt till det något udda begreppet i sammanhanget, nämligen <RECODABLE> i gränssnittet. [↑](#footnote-ref-1)
2. I outputformatet Intermediate, parameter out=Inter(1) ger kolumn med status. [↑](#footnote-ref-2)
3. Tillkommit som ett användarkrav och används inom ESS för *Foreign Trade Statistics*. [↑](#footnote-ref-3)
4. Vilket värde som de facto kan användas som bortfallsvärde är inte enkelt härledbart. Om annat bortfallsvärde används så editera metadatafilen [RDA] manuellt. Se τ-Argus-manualen. [↑](#footnote-ref-4)
5. ”A priori fil” förklaras till format och innehåll i τ-Argus-manualen. [↑](#footnote-ref-5)