This document is an historical remnant. It belongs to the collection Skeptron Web Archive (included in  
Donald Broady's archive) that mirrors parts of the public Skeptron web site as it appeared on 31 December 2019, containing material from the research group Sociology of Education and Culture (SEC) and the research programme Digital Literature (DL). The contents and file names are unchanged while character and layout encoding of older pages has been updated for technical reasons. Most links are dead. A number of documents of negligible historical interest as well as the collaborators’ personal pages are omitted. The site's internet address was since Summer 1993 www.nada.kth.se/~broady/ and since 2006 www.skeptron.uu.se/broady/sec/.

Introduktion till SPSS 10.0

version 1.3

2002-10-21

Mikael Börjesson

ILU och Pedagogiska institutionen, Uppsala universitet

[mikael.borjesson@ilu.uu.se](mailto:mikael.borjesson@ilu.uu.se)

vissa avsnitt av

Sverker Lundin

Matematiska vetenskaper, Chalmers tekniska högskola

[sverker@math.chalmers.se](mailto:sverker@math.chalmers.se)

Innehållsförteckning

[I. Inledning 4](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[I.1 Disposition och textförklaringar 4](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[I.2 Hur SPSS är uppbyggt 5](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[I.3 Några allmänna tips: 6](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[I.3.1 … lär dig använda kortkommandon 6](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[I.3.2 … spara en syntaxfil för varje dataset 7](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[I.3.3 ... utnyttja journalfilen 7](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[I.3.4 … Display Data Info och Utilities/File Info 7](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[I.3.5 ... utnyttja SPSS Syntax Guide, Help Topics och Tutorials 8](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[I.3.6 … skapa macron i Word 8](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[II. Generella inställningar 10](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[III. Dataredigeraren 12](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[III.1 Dataarket 12](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[III.1.1 De olika menyernas funktioner – en översikt 13](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[III.2 Variabelarket 14](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[III.2.1 Redigering i variabelarket 16](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[III.3 Att öppna och spara filer 17](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IV. Frekvenstabeller 19](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[V. Korstabeller 22](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[V.1.1 Korstabell med fler än två variabler 26](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[V.1.2 Något om olika inställningar och statistiska mått i samband med korstabeller 29](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[VI. Jämförelse av medelvärden 31](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[VII. Multipla svar-tabeller 33](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[VIII. Generella tabeller 36](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX. Omkodning, beräkningar, etc. av variabler 38](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.1 Omkodning av samma variabel... 38](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.1.1 ... via menyn 38](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.1.2 ... via skript 40](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2 Omkodning, beräkningar, etc. av en variabel till en annan variabel 42](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.1 Placering av ny variabel 42](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.2 Aggregation av variabel 42](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.3 Användning av Categorize Variable för att automatiskt koda om en variabel i lika stora grupper 45](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.4 Skapande av kombinerad variabel 46](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.5 Beräkningar av variabler 49](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.6 Räkna antal specifika värden i ett antal variabler 50](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.7 Omkodning av en numerisk variabel till en strängvariabel och vice versa 51](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.8 Automatisk omkodning av strängvariabel till numerisk variabel 53](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.9 Automatisk koncentration värdena i en numerisk variabel 54](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.10 Skapande av kombinerade strängvariabler 55](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.11 Några tips om omkodning och konstruktion av ett dataset 56](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[IX.2.12 Några tips för att koda om SPSS-data för att passa korrespondensanalyser i SPAD 57](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[X. Filter 59](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XI. Etiketter på värden 61](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XII. Användande av Word och Excel för skapande av skript 62](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XII.1 Att skapa etiketter 62](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XII.2 Att omkoda från numerisk variabel till strängvariabel 65](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XII.3 Att omkoda strängvariabler till numeriska variabler och använda värden i strängvariabeln till etiketter i den numeriska variabeln 68](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XIII. Aggregering och transponering av dataset 71](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XIII.1 Aggregering 71](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XIII.1.1 Exempel 1 71](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XIII.1.2 Exempel 2 72](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XIII.2 Transponering av data 74](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XIV. Automatisk filtrering efter ett antal kriterier 75](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XIV.1 Genomgång av kommandon 76](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XV. Sammanslagning av flera dataset och skapande av deldataset 81](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XV.1 Sammanslagning av dataset 81](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XV.1.1 Att lägga till variabler från ett annat dataset 82](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XV.1.2 Att lägga till individer från ett annat dataset 84](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XV.2 Att skapa deldataset 85](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XV.3 Att skapa dataset där individerna utgör endast en rad 85](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XVI. Att läsa in textfiler (och därmed skapa nya dataset) 92](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

[XVII. Att koda data för överföring till SPAD (program för bland annat korrespondensanalys) 99](file:///C:\a-skeptron-lokal\broady\arkiv\a\skeptron\p-borjesson-021121-spss-intro.doc)

# Inledning

Denna introduktion har tillkommit som ett försök att systematisera den kunskap som erhållits genom några års osystematiska utforskningar av statistikprogrammet SPSS möjligheter och begränsningar. Tanken är att den skall kunna vara en guide för såväl nybörjare som mer vana användare. En möjligen omöjlig ambition.[[1]](#footnote-1) Introduktionen kan dock användas på olika sätt. Man kan naturligtvis läsa den i den ordning avsnitten kommer (rekommenderas för den som inte tidigare arbetat med SPSS), men det går också bra att bara läsa de avsnitt man är intresserade av för stunden (vissa avsnitt förutsätter dock kunskaper och färdigheter som presenterats i andra avsnitt, där så är fallet har jag försökt att tydligt markera detta genom hänvisningar till andra nödvändiga avsnitt). En introduktion av detta slag kan inte bli fullständig. Jag har valt att fokusera på de moment som jag anser viktigast för att kunna hantera stora dataregister, göra analyser av dessa och anpassa dataset för korrespondensanalys. Introduktionen är praktiskt hållen, jag har försökt att ge olika tillvägagångssätt för procedurerna – den som ämnar arbeta mycket med statistik bör lära sig att själv skriva skript medan nybörjaren som ej har för avsikt att arbeta mycket med statistik kan utföra allt genom att klicka sig fram via menyer. Det finns mycket lite om statistiska beräkningar i det följande. Det som återfinns har jag hämtat från en utförlig och överskådlig grundbok i statistik, Karin Dahmström, *Från datainsamling till rapport,[[2]](#footnote-2)* som rekommenderas för den som vill arbeta med statistik.

## Disposition och textförklaringar

En introduktion kan skrivas på många sätt. Jag har här valt att försöka eftersträva en logisk arbetsprocess där omkodning av data har fått en central roll. Ambitionen är att läsaren på egen hand skall kunna bearbeta ett dataset så att det sedan blir möjligt att göra analyser såväl i SPSS som i SPAD (program för korrespondensanalys).

I kapitlen I Inledning och II Generella inställningar ges en översikt över hur SPSS är uppbyggt, information om hur man ställer in grundläggande inställningar samt ett antal generella råd för arbetet med statistik.

Kapitel III Dataredigeraren redogör för hur dataredigeraren med dess två ark, dataarket och variabelarket är uppbyggda och hur dessa kan användas.

Kapitel IV Frekvenstabeller och V Korstabeller går igenom hur man gör frekvenstabeller och korstabeller, vilket är en förutsättning för att kunna gå vidare med det för denna introduktion centrala avsnittet: omkodning av variabler. I kapitel VI Jämförelse av medelvärden redogörs för hur man jämför olika variabler med avseende på medelvärden och andra spridningsmått. Hur man skapar tabeller som summerar informationen i ett stort antal variabler visas i kapitel VII Multipla svar-tabeller. För att skapa en bra översikt över en viss variabels karakteristika kan man göra generella tabeller; detta redovisas i kapitel VIII Generella tabeller.

Hur man kodar om variabler och skapar nya variabler diskuteras grundligt i kapitel IX Omkodning, beräkningar, etc. av variabler. Därtill ges också en rad förslag på när det är lämpligt att skapa nya variabler. En användning är att skapa filter för att kunna analysera delar av ett omfattande material. Hur filter skapas och används gås igenom i kapitel X Filter.

När man skapar nya variabler är det vanligtvis praktiskt att sätta etiketter för de olika värdena för att det skall bli möjligt att enkelt analysera data. Kapitel XI Etiketter på värden ägnas åt detta.

För att underlätta arbete med att skapa skript som talar om hur exempelvis variabler ska kodas om och hur etiketter skall sättas ut kan man med fördel använda sig av Word och Excel, vilket redogörs för i kapitel XII Användande av Word och Excel för skapande av skript.

I kapitel Aggregering och transponering av dataset visas hur man kan skapa aggregerade dataset och hur man kan transponera dataset, dvs. hur man byter plats på variabler och observationer. Kapitel XIV Automatisk filtrering efter ett antal kriterier ges ett exempel på hur man genom ett antal kommandon kan skapa ett automatiskt filter för att selektera vissa variabler.

Kapitel XV Sammanslagning av flera dataset och skapande av deldataset och XVI Att läsa in textfiler (och därmed skapa nya dataset) diskuteras hur man går till väga för att behandla olika dataset och på olika sätt skapa nya dataset.

Slutligen ges i kapitel XVII Att koda data för överföring till SPAD (program för bland annat korrespondensanalys) en vägledning till hur data måste kodas för att det skall gå att importera datasetet i SPAD.

Några ord behöver också sägas om layouten. I det följande kommer jag att använda några olika stilar för att markera följande:

* Kortkommandon skrivs mellan < och >, t.ex. <ctrl+s>.
* Programskript är skriven med teckensnittet Arial 10 p. Utbytbar text (framför allt variabler) i programskript skrivs med fet stil.
* Originaltermer för procedurer i SPSS skrivs med kursiv, exempelvis *Data Editor*.
* Menyhänvisningar skrivs separerade med / i hierarkisk ordning, exempelvis *File/Open/Data*.

## Hur SPSS är uppbyggt

SPSS omfattar av tre olika huvudfönster:

1. Dataredigeraren (*Data Editor*) som har två ark själva databasen (*data view*) och variabelarket (*variable view*). Alla databaser är uppbyggd som en tabell där raderna utgör observationerna och kolumnerna variablerna. (Se kapitel III Dataredigeraren)
2. Resultatfönstret (*Output view*) där alla resultat (tabeller, diagram etc.) presenteras. Varningar och felmeddelanden visas också här.
3. Syntaxredigeraren (*Syntax Editor*) där man skriver sina kommandon och kör dessa. (Se avsitt IX.1.2 ... via skript)

En ytterligare central funktion är journalfilen, där alla kommandon sparas.

Det finns ett antal olika filformat i SPSS:

Dataseten har filbeteckningen ”*.sav*”.

Syntaxfiler har filbeteckningen ”*.sps*”.

Outputfiler har filbeteckningen ”*.spo*”.

Journalfilen har filbeteckningen ”*.jnl*”.

## Några allmänna tips:

Även om det bara handlar om att göra en statistisk undersökning för en C-uppsats finns mycket tid att vinna på att använda sig av några enkla sätt för att underlätta arbetet. Kortkommandon gör arbetet definitivt snabbare och dessa har man ofta nytta av oavsett vilket program man använder. Samma sak gäller macron, små program som utför olika rutiner i Word och Excel. Det är också viktigt att noga dokumentera det man gör. Därför rekommenderar jag att man skapar en syntaxfil för varje dataset. I syntaxfilen samlar man alla de kommandon man genomför. Det är viktigt att spara dessa filer för att sedan kunna gå tillbaka för att se hur man exempelvis genomfört vissa omkodningar. (En annan fördel med syntaxfilerna är att man kan använda dem för att genomföra samma bearbetningar av andra likartade dataset.) Därtill finns även journalfilen där nästintill alla kommandon sparas. Är man osäker på hur ett kommando skrivs kan man söka i *SPSS Syntax Guide*. Eftersom dataseten snabbt blir omfångsrika är det svårt att få en överblick över dem. Man kan då använda sig av *Display Data Info* för att få en översikt över datasetet.

### … lär dig använda kortkommandon

Om du inte redan använder kortkommandon vill jag rekommendera att du börjar med det. Det gör att arbetet går mycket snabbare. Dessutom undviker du förhoppningsvis att få musarm. De viktigaste kortkommandona är:

|  |  |  |  |
| --- | --- | --- | --- |
| Typ of Command | Command Name | Modifiers | Key |
| Edit | Cut | Ctrl+ | X |
|  | Copy | Ctrl+ | C |
|  | Paste | Ctrl+ | V |
|  | Select All | Ctrl+ | A |
|  | Find | Ctrl+ | F |
|  | Replace | Ctrl+ | H |
|  | Undo | Ctrl+ | Z |
|  |  |  |  |
| File | Save | Ctrl+ | S |
|  | Open | Ctrl+ | O |
|  | Print | Ctrl+ | P |
|  |  |  |  |
| Insert | Page Break | Ctrl+ | Return |
|  | Footnote Now | Alt+Ctrl+ | F |
|  |  |  |  |
| Förflyttning i dokument | Para Down | Ctrl+ | Down |
|  | Para Up | Ctrl+ | Up |
|  | End of Document | Ctrl+ | End |
|  | Word Left | Ctrl+ | Left |
|  | Word Right | Ctrl+ | Right |
|  |  |  |  |
| Växla mellan program/fönster | Change Window | Alt+ | Tab |

Ett annat sätt att underlätta arbetet är att använda musens högerknapp, dvs. ”att högerklicka”. Beroende på var markören är placerad kommer det då upp ett antal alternativ som du kan välja mellan.

### … spara en syntaxfil för varje dataset

För att underlätta arbetet i SPSS är det viktigt att noga dokumentera det man gör. Jag har som vana att spara alla de skript jag gör i en speciell syntaxfil. Enklast är att spara en syntaxfil för varje dataset och att döpa syntaxfilerna efter datasetens namn. Att använda syntaxfönstret kan spara mycket tid, dels om man gjort något misstag och måste göra om ett antal operationer, dels om man skall göra ungefär samma sak med flera olika dataset.

Vissa registermanipulationer sker i ett antal steg som alltid måste göras i samma ordning. I dessa behöver man inte utföra ett kommando i taget, utan kan markera allt som skall göras och köra det i ett svep.

I syntaxfilen går det att skriva in förklaringar i texten. Man börjar då med att skriva \* och programmet betraktar då all text som skrivs på samma rad som dold. Raden måste avslutas med en punkt. Detta kan se ut som följer:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* För att öppna en fil skriver man GET FILE ='filens namn’. Se nedan.

GET

FILE='C:\Statistikfiler\gym9798\Dataset\kombinerade set\tot-IV-stud.sav'.

### ... utnyttja journalfilen

Ifall man av misstag tar bort en syntaxfil eller glömmer att trycka *Paste* när man gör en omkodning eller dylikt kan man alltid öppna journalfilen för att se hur man gått till väga och därmed återskapa en syntaxfil. I journalfilen samlas nämligen i princip alla de kommandon man genomför. Denna fil blir förvisso rätt så rörig om man jobbar med många olika dataset – därför är det bra att spara separata syntaxfiler – men man kan söka i filen genom att använda <ctrl+f>. Det framgår också av journalen när man öppnar ett nytt dataset.

### … Display Data Info och *Utilities/File Info*

När dataseten blir omfattande är det svårt att hålla reda på alla variabler och vilka värden de har. Ett enkelt sätt att få en överblick över sitt dataset är att skapa en textfil som beskriver datasetet. Detta gör man genom att gå in på *File/Display Data Info* och därefter välja det dataset man vill ha information om. Observera att detta dataset inte redan kan vara öppnat. Öppna i så fall ett nytt tomt dataset (*File/New/Data*) och välj sedan det datasetet du vill ha information. Du kan också gå in i syntaxredigeraren och skriva följande för att få en textfil över datasetet:

SYSFILE INFO

'C:\Statistikfiler\gym9798\Dataset\kombinerade set\tot-IV-stud.sav'.

Mellan de enkla citattecknen skriver du in ”adressen” till den datafil du vill ha information om. Textfilen över datasetet visas i resultatfönstret (*Output view*). Enklast är att dubbelklicka på texten, markera allt <ctrl+a>, kopiera <ctrl+c> och klistra in <ctrl+v> i ett Word-dokument.

Textfilen ser exempelvis ut enligt följande:

REG1O2 (reg1o2) ingår i gym97/98 6

Measurement level: Scale

Format: F1 Column Width: 6 Alignment: Right

Value Label

1 ingår i gym97/98

KON (kon) kön 7

Measurement level: Ordinal

Format: F1 Column Width: 4 Alignment: Right

Value Label

1 pojke

2 flicka

3 kön, ej uppgift

FODAR (fodar) födelseår 8

Measurement level: Ordinal

Format: F2 Column Width: 5 Alignment: Right

Här framgår att variabel 6 heter REG1O2 och att variabeln har etiketten (*label*)(reg1o2) ingår i gym97/98. Det finns ett värde (*value*) specificerat med en etikett (*label*), nämligen värdet 1 och etiketten ingår i gym97/98. Mer om variabler finns i avsnitt III.2 Variabelarket.

I princip samma resultat kan uppnås genom att man går in på *Utilities/File Info*, då erhåller man information om alla variablerna i det dataset man arbetar med.

### ... utnyttja SPSS Syntax Guide, Help Topics och Tutorials

Om man är osäker på hur kommandon skrivs eller vill undersöka hur olika procedurer kan skrivas i skriptform kan man med fördel ta del av den gedigna *Syntax Guide* som finns i programmet. Gå in på *Help/Syntax Guide/Base* och du får upp ett över 1 400 sidor långt dokument över hur man skriver skript i SPSS. Detta förutsätter dock att man har Acrobat Reader i sin dator. Programmet är närmast ett standardprogram i alla datorer och om det inte finns går det enkelt att ladda ner från nätet.

En annan användbar hjälpfunktion är *Help/Topics*  som fungerar som ett uppslagsverk över allt i SPSS. Det går att söka på enskilda ord samt att slå upp olika avsnitt som Using the Data Editor, Recoding Variables, och Aggregating Data.

Från *Topics* kan man även gå till *Tutorials* (alternativet är att gå direkt till *Help/Tutorials*) som är en lätt överskådlig handledning som visar steg för steg hur vissa saker utförs.

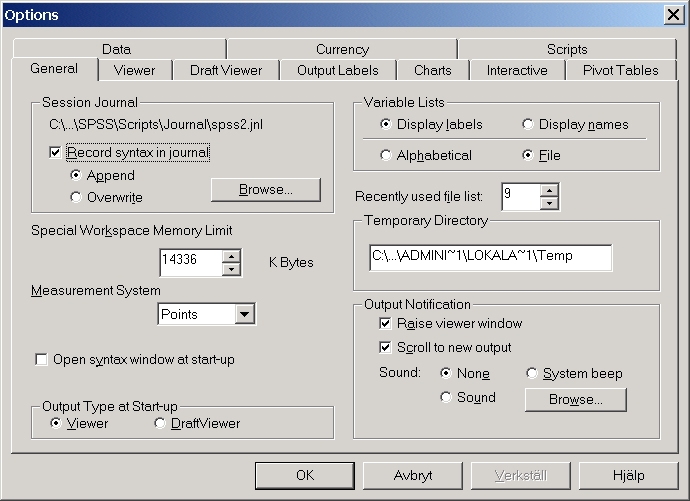
Prova gärna dessa olika funktioner redan nu! Och kom också ihåg att dessa hjälpfunktioner finns, även om man tycker att man behärskar programmet finns det mycket mer att lära och hjälpfunktionerna är ofta ett enkelt sätt att gå vidare.

### … skapa macron i Word

Nedan – i kapitel XII Användande av Word och Excel för skapande av skript – kommer vi att gå igenom hur man med hjälp av Word kan redigera skript. För att underlätta detta kan det vara värt mödan att skapa några macron (program i Word som automatiskt gör ett antal specificerade manövrer). Att skapa ett nytt macro görs genom att man går in på *Tools/Macro/Recode New Macro*.

# Generella inställningar

Ett första steg innan vi börjar med databehandlingen är att ställa in några grundläggande och generella inställningar. Detta görs via *Edit/Options*. Följande fönster visas:



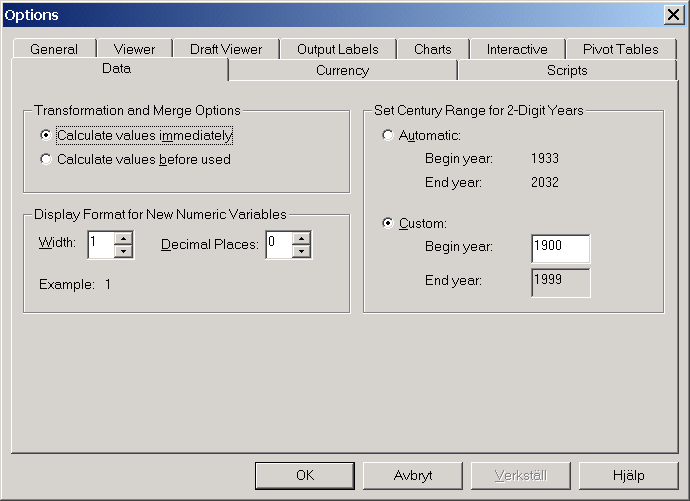
Alla centrala kommandon som du som användare gör i SPSS registreras i en journalfil. Detta gör att man kan i efterhand gå in och se vad man gjort vid ett tidigare skede (hur man exempelvis kodat en viss variabel). Man kan också bakvägen se hur vissa kommandon skrivs och kopiera dessa för att använda dem igen. Det är därför viktigt att man kryssar för *Record syntax in journal* och att man kryssar för *Append* och **inte** *Overwrite* (man vill att nya kommandon skall läggas till de gamla och inte skriva över dessa). Vill man ändra platsen på datorn för journalen eller döpa om journalen så klickar man på *Browse…* Jobbar man ofta med SPSS kan man exempelvis döpa journalen efter tidsperiod, exempelvis ”Journal-2002.jnl”. Arbetar man på flera datorer är det bra att döpa journalen utifrån efter dator, exempelvis ”Journal-2002-stationär.jnl” och ”Journal-2002-bärbar.jnl”.

I många av SPSS menyfunktioner ges en lista med variabler att välja från. Denna lista – se under avdelning *Variable list* – kan baseras på variablernas namn eller etiketter (*labels*). Jag föredrar etiketter eftersom dessa innehåller mer information om variabeln och gör det lättare att få rätt variabel. Man kan även välja hur man vill visa variablerna, alfabetiskt eller enligt den position de har i datasetet. Personligen prefererar jag att utgå från deras placering i datasetet eftersom detta förhoppningsvis följer en viss logik.

Arbetsminnet (*Special Workspace Memory Limit*) brukar jag ställa till ca 14000 kbs (jag har inte lyckats få en högre siffra, programmet säger av någon outgrundlig anledning ifrån oavsett om datorns internminne överstiger gränsen betydligt.

Slutligen sätter jag antalet nyligen använda filer till 9 eftersom det ofta är enkelt att gå in och öppna en fil via denna funktion.

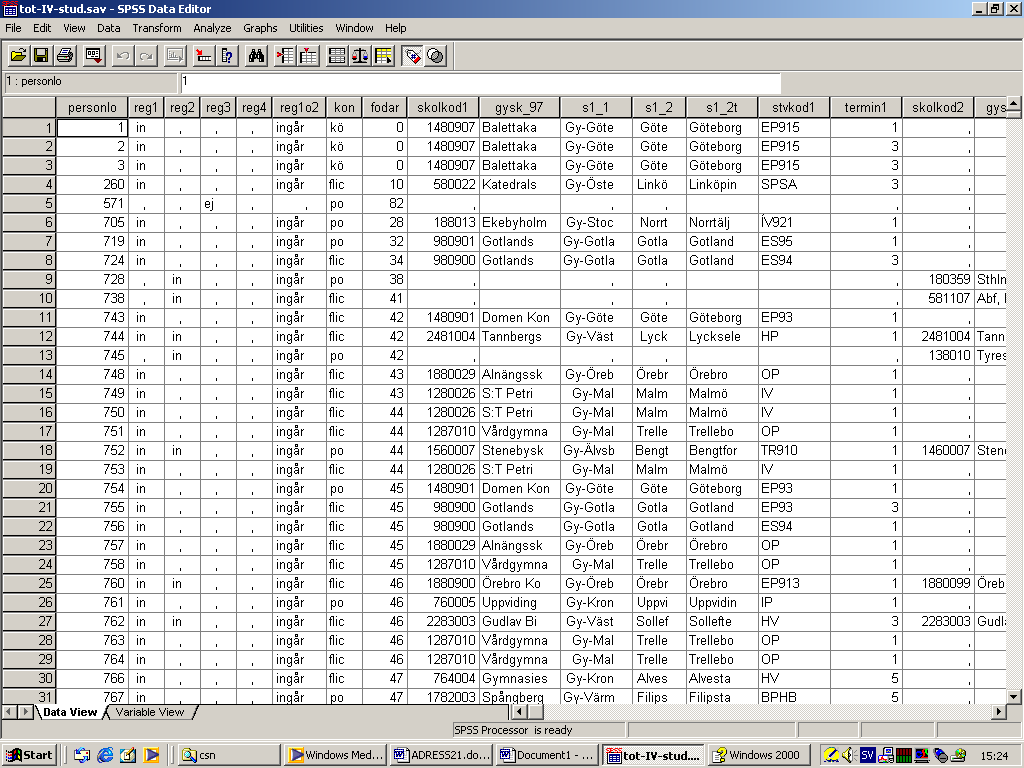
Andra vettiga inställningar att justera är under fliken Data, där jag under *Display Format for New Numeric Variables* ställer *Width* till 1 och *Deciamal places* till 0. Detta för att få bort onödiga decimaler och onödigt många tecken.



# Dataredigeraren

SPSS har två fönster i dataredigeraren – *data editor* – (detta gäller från och med SPSS 10.0). Det ena – *data view* – är själva databasen. Det andra – *variable view* – visar alla variablerna och information om dessa. För att byta mellan de två fönstren klickar man antingen längst ned till vänster på de olika arken eller använder <ctrl+t>.

## Dataarket



Dataarket är så uppbyggt att raderna motsvarar individer (eller andra former av observationer, exempelvis data för enskilda skolor eller utbildningar). Kolumnerna utgör olika variabler. I fallet ovan, hämtat från våra gymnasieregister som totalt omfattar 425 000 individer, ser vi de 31 första individerna och ett antal variabler. Den första variabeln, personlp, är individernas löpnummer. Detta är samma i alla register vilket gör att man kan matcha information om individerna från olika register och därmed skapa nya register.

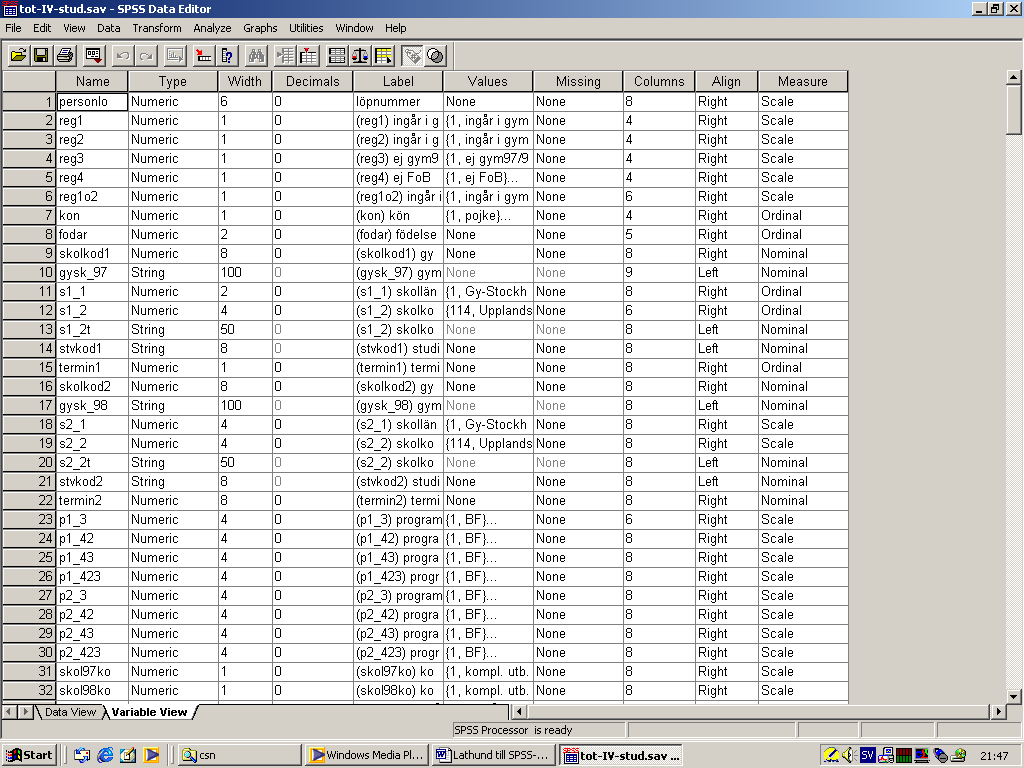
### De olika menyernas funktioner – en översikt

Som i alla windowsprogram finns ett antal menyer att välja på. Vi ska här kort beskriva hur de är strukturerade. Det finns i *SPSS Dataview* tio olika huvudmenyer och de har följande funktionsområden:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Huvudmeny** | **Kortkom.** | **Huvudområde** | **Centrala funktioner** | **Beteckning** | **Se vidare** |
| **File** | Alt+f | Filhantering | Öppnar filer | Open | III.3 |
|  |  |  | Sparar filer | Save | III.3 |
|  |  |  | Läser in textfiler | Read Text Data | XVI |
| **Edit** | Alt+e | Redigering | Klipp ut, klistra in, ångra, sök, ersätt, mm. | Cut, Copy, Undo, Find, Replace, etc. |  |
| **View** | Alt+v | Utseende | Justeringar av programmets vertygsfält |  |  |
| **Data** | Alt+d | Datahantering | Insättning av variabler | Insert Variable | III.2.1.1 |
|  |  |  | Sorterar variabler | Sort Cases | XV.1.1.1 |
|  |  |  | Transponering | Transpose | XIII |
|  |  |  | Sammanslagning av filer | Merge Files | XV |
|  |  |  | Aggregering av dataset | Aggregate | XIII |
|  |  |  | Filtrering av dataset | Select Cases | X |
| **Transform** | Alt+t | Omkodningar och bearbetningar av data | Beräkna | Compute | IX.2.5 |
|  |  |  | Räkna | Count | IX.2.6 |
|  |  |  | Koda om i samma variabel | Recode into same variable | IX.1 |
|  |  |  | Koda om till annan variabel | Recode into different variable | IX.2 |
|  |  |  | Kategorisering av variabel | Categorize Variables | IX.2.3 |
|  |  |  | Automatisk omkodning | Automatic Recode | IX.2.8, IX.2.9 |
| **Analyze** | Alt+a | Statistiska analyser | Beskr. stat., frekvenstab. | Descriptive Stat./Frequencies | IV |
|  |  |  | Beskr. stat., korstab. | Descriptive Stat./Crosstabs | V |
|  |  |  | Generella tabeller | Custom Tables/General | VIII |
|  |  |  | Jämförelse medelvärden | Compare Means | VI |
|  |  |  | Multipla svar-tabeller | Multiple Responces-Tables | VII |
| **Graphs** | Alt+g | Grafiska presentationer | Olika typer av diagram (stapel, cirkel, linje, etc.) | Bar, Pie, Line, Histogram |  |
| **Utilities** | Alt+u | Nyttiga funktioner | Filinformation | File Info | I.3.4 |
|  |  |  | Definition av dataset | Define Set |  |
| **Window** | Alt+w | Fönster | Byta fönster |  |  |
| **Help** | Alt+h | Hjälp | Ämnen | Topics | I.3.5 |
|  |  |  | Handledning | Tutorial | I.3.5 |
|  |  |  | Syntaxguide | Syntax Guide | I.3.5 |

När man använder menyerna kan man med fördel använda kortkommandon. Genom att trycka <alt+h> får man upp menyn för *Help*. Genom att sedan endast ange den bokstav som står understruken i en viss funktion aktiveras den funktionen. Ange p efter <alt+h> så får du automatiskt upp *Help/Topics*.

## Variabelarket



Variabelarket listar alla variabler i datasetet. Fördelen med detta ark är att man relativt enkelt kan redigera variablerna. De olika kolumnerna avser följande:

* *Name* visar variabelns namn. Detta får endast ha 8 tecken. Man bör undvika åäö. Vissa tecken som punkt och bindestreck går ej att använda och ej heller att börja med en siffra (däremot kan man ha siffror på andra platser i namnet, vilket är vanligt förekommande och ibland praktiskt när det exempelvis gäller årtal och andra tal).
* *Type* anger vilken typ av variabel det är. De vanligaste är *numeric* (numerisk variabel) och *string* (strängvariabel). Numeriska variabler är som namnet antyder siffervariabler och här kan endast siffror förekomma. Strängvariabel är en ren textvariabel och här kan alla typer av tecken förekomma, dvs. även siffror.
* *Width* anger hur många tecken/siffror som förekommer i variabeln. Max för numeriska variabler är 40 siffror och för strängvariabler 255 tecken. (Defaultvärde kan ändras i Edit/Options/Data.)
* *Decimal* anger antalet decimaler för numeriska variabler. (Defaultvärde kan ändras i Edit/Options/Data.)
* *Label* beskriver variabeln i text (max 255 tecken). Tänk på att ha unika beskrivningar så att man i tabeller kan se vilken variabel det är frågan om. (Om man angett en label är det den som exempelvis framkommer i tabellutskrifter. (Ett tips är att även inkludera variabelns namn i beteckningen så håller man lättare koll på sina data.)
* *Values* anger vad olika värden betyder. Denna funktion används vanligtvis för numeriska variabler där exempelvis värdet 1 står för kvinna och värdet 2 för man. Genom att trycka på ikonen Value labels i verktygsfältet kan man visa vad de olika siffrorna står för (alternativt ta bort beskrivningarna och istället visa siffrorna). Funktionen kan också användas för strängvariabler där exempelvis k = ”kvinna” och m = ”man”.
* *Missing* anger vilka värden man satt som saknade värden. Vanligtvis sätts värde 0, 9, 99, 999 som missing.
* *Columns* anger med vilken vidd som variablerna (kolumnerna) visas i dataarket. Observera att detta värde inte behöver överensstämma med *Width*, man kan ha en *Width* på 255 men endast visa 10 tecken i databasen.
* *Align* talar om åt vilken sida texten är placerad, per default är strängvariabler vänsterställda och numeriska variabler högerställda.
* *Measure* anger hur de olika värdena i en variabel förhåller sig till varandra (variablerna delas in efter datanivå eller skaltyp).[[3]](#footnote-3) Detta har betydelse för vilka typer av statistiska analysmetoder som är möjliga att tillämpa. Dock spelar det ingen roll när det gäller korstabeller eller korrespondensanalys. (Jag brukar sällan bry mig om att ställa in rätt datanivå, men som sagt nödvändigt för att göra vissa typer av statistiska analyser.)

De bestämningar av variablerna som är viktigast är *name*, *type*, *width*, *labels* och *values*.

**När du bestämmer namn på en variabel tänk på att:**

1. börja med så korta namn som möjligt; man behöver sedan utrymme för att lägga till bokstäver/siffror/tecken när man skapar nya varianter av variablerna.
2. försök att namnge variablerna logiskt så att exempelvis alla variabler som rör skolor börjar med s och att när samma variabel förekommer för flera år namnge dem så att man enkelt kan särskilja de olika åren exempelvis s97, s98, s99.
3. bruka gärna understreck \_ för att separera olika information i variablerna, exempelvis kan döpa en variabel som kombinerar skola och program till s97\_p1 (observera att det inte går att använda bindestreck eller punkt.
4. i vissa fall kan det vara bra att ange källan för en viss variabel, såsom Folk och bostadsräkningen 1990, och vem informationen avser, exempelvis alla variabler om fadern från FoB-90 kan börja med f90 och alla variabler om modern m90.

**När du bestämmer eller specificerar värden på en variabel tänk på att:**

1. Göra beskrivningarna unika, dvs. använd exempelvis inte endast ”ja” eller ”nej” utan specificera också vilken variabel det är frågan om (”Positiv-EU: ja” och ”Positiv-EU: nej”). Detta är framför allt viktigt när man tänker sig göra korrespondensanalys då det är viktigt att kunna skilja de olika värdena åt (tio variabler uppdelade på ”ja”, ”nej” och ”vet inte” blir omöjliga att tolka i en graf!).
2. Om du avser använda korrespondensanalys tänk på att endast 20 tecken när du specificerar värdena eftersom SPAD endast tar med de första 20 tecknen när man konverterar en fil från SPSS till SPAD (det går att redigera värdebeteckningarna i SPAD, men är onödigt krångligt).

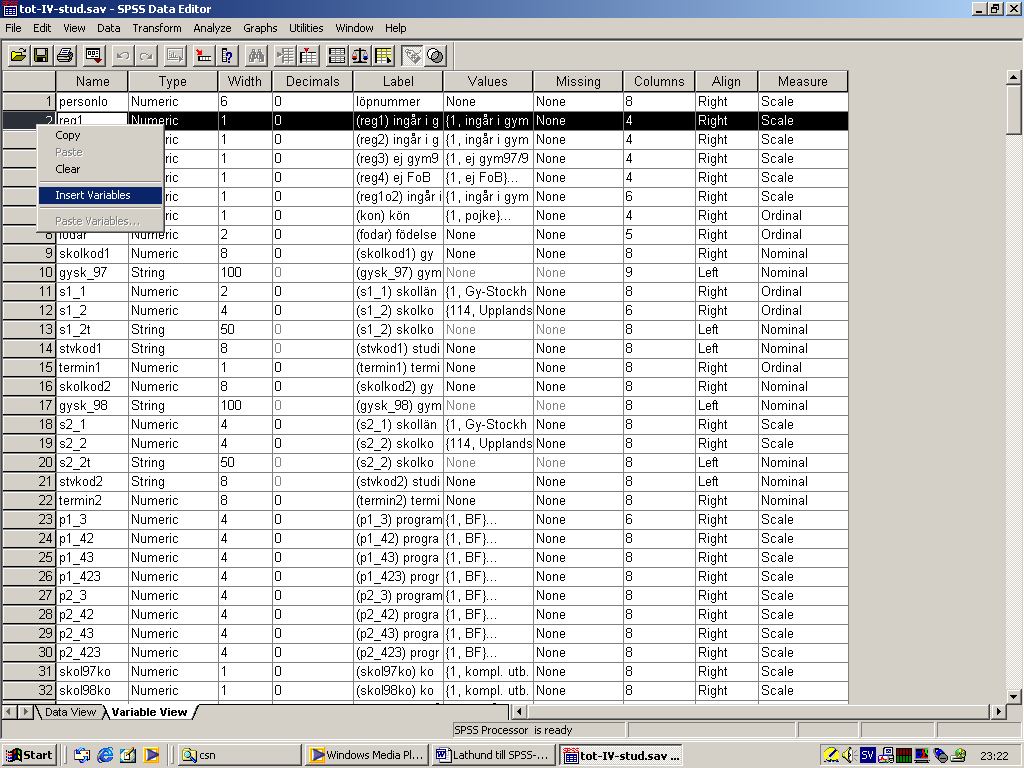
### Redigering i variabelarket

Det går att relativt enkelt redigera informationen om variablerna i variabelarket. Exempelvis kan man kopiera värdet för *width* från en variabel och klippa in detta på en eller flera andra variabler. Det går även att kopiera ett antal olika värden för *width* och klippa in dessa för andra variabler (observera att det måste vara samma antal variabler).

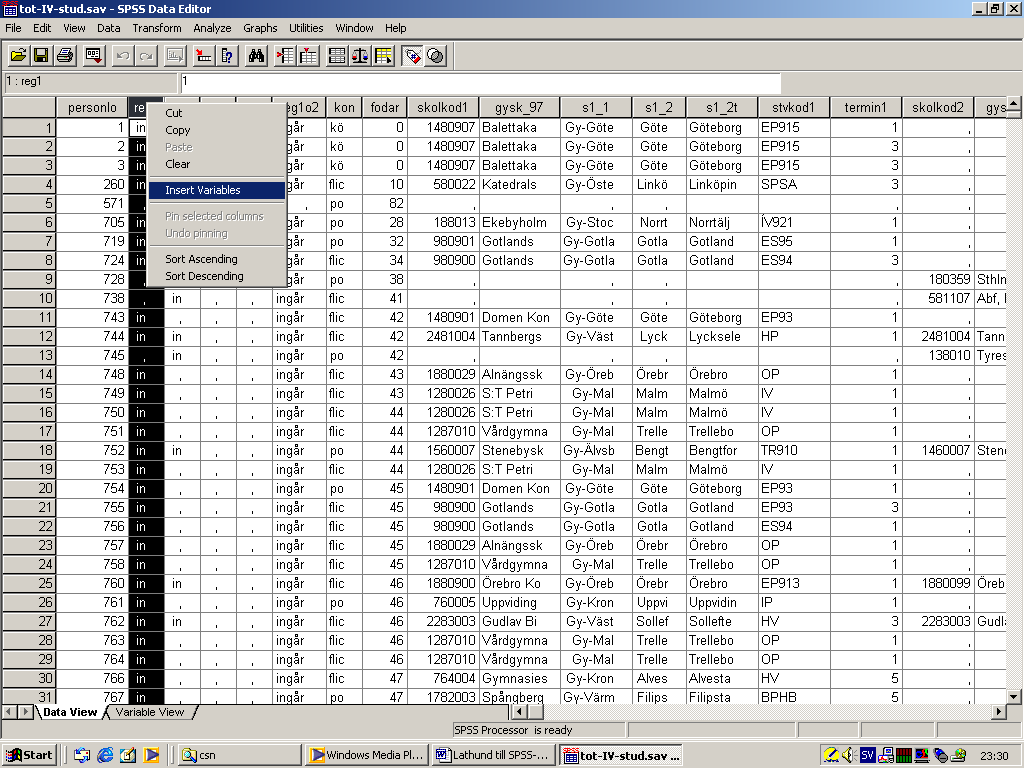
#### Infogande av ny variabel

För att foga in en ny variabel kan man gå tillväga på flera sätt:

1) genom variabelarket. Markera hela variabeln genom att föra markören till den variabel man vill sätta in den nya variabeln före. Ställ markören till vänster (vid numret för variabeln) och högerklicka. Välj *Insert Variables*.



2) genom dataarket Markera hela variabeln genom att föra markören till den variabel man vill sätta in den nya variabeln före. Ställ markören ovanför (vid namnet för variabeln) och högerklicka. Välj *Insert Variables*.



3) genom att omkoda en variabel till en ny variabel (se IX.2 Omkodning, beräkningar, etc. av en variabel till en annan variabel). Då hamnar den nya variabeln automatiskt sist i arket. Variabeln går sedan att flytta genom att markera och dra eller genom att klippa ut och klistra in.

#### Borttagning av en variabel

Det går även att ta bort en variabel. Markera den variabel som skall tas bort och högerklicka. Välj *Clear*.

## Att öppna och spara filer

För att öppna en redan befintlig fil gå in på *File/Open/Data*. Vill du öppna ett nytt dataark, gå in på *File/New/Data*. För att öppna en redan befintlig fil via syntaxredigeraren, skriv

GET

FILE='C:\Statistikfiler\gym9798\Dataset\kombinerade set\tot-IV-stud.sav'.

För att spara en ny fil gå in på *File/Save as*. För att spara den fil du arbetar med använd <ctrl+s> eller gå in på gå in på *File/Save*. För att spara en fil via syntaxredigeraren, skriv

SAVE OUTFILE='C:\Statistikfiler\gym9798\Dataset\kombinerade set\tot-IV-stud.sav'

/COMPRESSED.

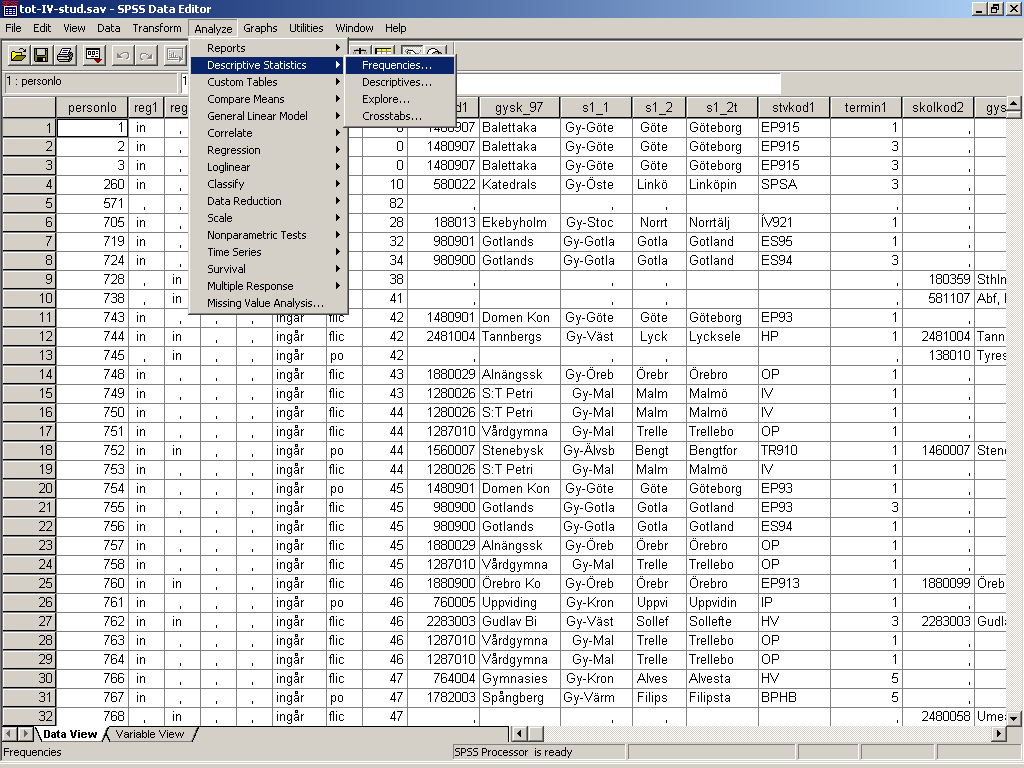
Vill du spara en ny fil eller redan befintlig fil i den folder som de redan arbetar i räcker det att skriva:

SAVE OUTFILE='Tot-IV-stud.sav'

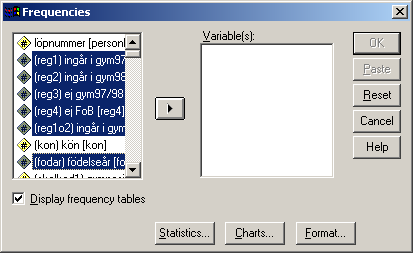
/COMPRESSED.

# Frekvenstabeller

Ett första steg för att analysera sina data är att dra ut frekvenstabeller över variablerna. Detta är enkelt gjort genom menyn, välj *Analyze/Descriptive Statistics/Frequencies...* .



Specificera sedan vilka variabler du vill få frekvenstabeller för genom att markera dem i rutan till vänster och sedan föra över dem till rutan till höger *Variable(s)* genom att klicka på piltangenten. Vill man ändra sig markerar man en variabel i den högra rutan och klickar bort den med piltangenten (Som nu vänt riktning). Man bör undvika att få med löpnummer eftersom detta ofta inte är av intresse och leder till alltför omfattande tabeller.[[4]](#footnote-4) Man kan välja upp till 500 variabler på en gång. För att välja olika variabler som ej ligger efter varandra i ordningen kan man hålla ned *Ctrl*-tangentensamtidigt som man markerar variablerna. Ibland kan det bli problem med att dra ut alltför omfattande tabeller. Det går dock att ”lura” programmet genom att först specificera en variabel med få värden, exempelvis kön, och sedan välja variabeln med många värden.



När man väl valt ut variabler kan man specificera andra önskemål. Genom att klicka på *Statistics...* kan man välja dela in materialet efter *Percentile Values* i:

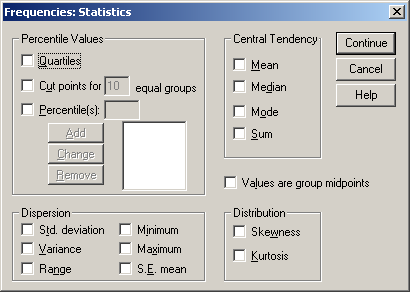
*Quartiles –* kvartiler (delas upp i fyra delar, 25 % i varje)

*Cut point for 10 equal groups* – deciler (delas upp i 10 lika stora grupper, man även välja att dela upp i exempelvis 2, 20, 50 lika stora grupper)

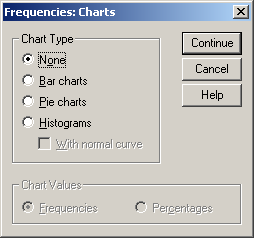
*Percentlies* – percentiler (specificerar vilken percentil[[5]](#footnote-5) man vill dela in materialet efter. Här kan man välja flera olika percentiler.

Detta är användbart när man t.ex. vill aggregera variabler med många värden såsom inkomst som förslagsvis kan delas in i tio deciler (dvs. tio lika stora grupper), se nedan.[[6]](#footnote-6)

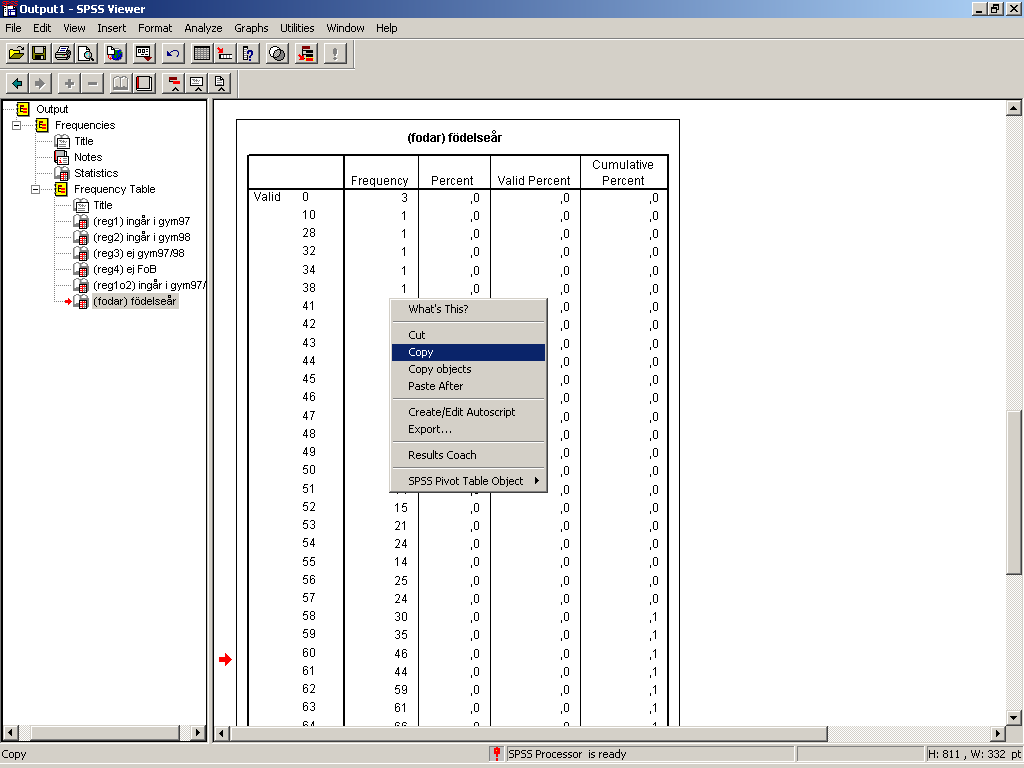
Man kan även få *mean* (medelvärde), *median* (median), *mode* (?)och *sum* (summa). Därtill kan det vara bra att få mått på spridningen (*dispersion*) i variabeln. Här kan man välja *Std. deviation* (standardavvikelsen), *Variance* (variansen), *Minimun*, *Maximum*, *Range* (variationsvidden – skillnaden mellan max- och min-värde), och *S.E. mean* (medelvärdets standardfel). För att erhålla ett mått på hur distributionen är fördelad kring centralvärdet kan man välja *Skewness* och *Kurtosis*.



Man kan vidare välja att visa olika former av diagram: *Bar charts* (stapeldiagram), *Pie charts* (cirkeldiagram) och *Histograms* (histogram). Det senare förutsätter att man använder kontinuerliga variabler.



Frekvenstabellerna presenteras i resultatfönstret – *Output*. Genom att markera en frekvenstabell (antingen i trädet till vänster eller i själva tabellen till höger) och högerklicka kan man välja att kopiera tabellen (det går naturligtvis bra att använda <ctrl+c>). Därefter kan tabellen klippas in i ett Excel-ark (Se vidare avsnitt för hur Excel-arken kan användas för omkodningar av variabler).



# Korstabeller

Ett mycket användbart analysverktyg är korstabellen. Korstabellen kan sägas vara en beskrivning av hur en variabels värden är fördelade på en annan variabels värden. Exempelvis kan vi analysera hur variabeln kön är fördelad på gymnasieprogram. Vi väljer då *Analyze/Descriptive Statistics/Crosstabs...*. Som rader (*rows*) sätter vi in variabel p2\_3 och som kolumner (*columns*) variabeln kon.



När vi trycker på *Paste* får vi följande skript:

CROSSTABS

/TABLES=**p2\_3** BY **kon**

/FORMAT= AVALUE TABLES

/CELLS= COUNT .

Resultatet i *Output*-fönstret inleds med en summering (*Case Processing Summary*):

Case Processing Summary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Cases |  |  |  |  |  |  |
|  | Valid |  | Missing |  | Total |  |  |
|  | N | Percent | N | Percent | N | Percent |  |
| (p2\_3) program 98 \* (kon) kön | 309141 | 100,0% | 0 | ,0% | 309141 | 100,0% |  |

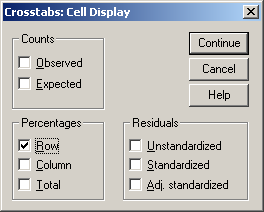
I det här fallet har jag filtrerat datasetet efter individer som finns i gymnasieskolan 1998 (de individer som endast gått i gymnasiet 1997 och finns med i det totala registret utesluts i analysen). I och med att vi har uppgifter om program för alla individer och om kön (för några individer saknas uppgift men de har fått värdet Ej uppgift) så får vi 100 % giltiga fall. Själva korstabellen ser ut enligt följande:

(p2\_3) program 98 \* (kon) kön Crosstabulation

Count

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | (kon) kön |  |  | Total |  |
|  |  | pojke | flicka | kön, ej uppgift |  |  |
| (p2\_3) program 98 | BF | 4001 | 11722 | 21 | 15744 |  |
|  | BP | 5754 | 132 | 3 | 5889 |  |
|  | EC | 12907 | 235 | 4 | 13146 |  |
|  | EN | 2323 | 60 | 2 | 2385 |  |
|  | ES | 4919 | 10226 | 25 | 15170 |  |
|  | FP | 11070 | 313 | 5 | 11388 |  |
|  | HP | 7082 | 7124 | 14 | 14220 |  |
|  | HV | 591 | 3426 | 8 | 4025 |  |
|  | HR | 6441 | 7861 | 5 | 14307 |  |
|  | IP | 6533 | 425 | 3 | 6961 |  |
|  | LP | 792 | 1171 | 3 | 1966 |  |
|  | MP | 5206 | 5646 | 12 | 10864 |  |
|  | NP | 3061 | 3831 | 12 | 6904 |  |
|  | NV | 36936 | 25525 | 68 | 62529 |  |
|  | OP | 1436 | 8517 | 9 | 9962 |  |
|  | SP | 29129 | 48243 | 131 | 77503 |  |
|  | IB | 353 | 615 | 21 | 989 |  |
|  | IV | 9190 | 7084 | 63 | 16337 |  |
|  | IVIK | 666 | 653 | 85 | 1404 |  |
|  | SM | 9414 | 6263 | 44 | 15721 |  |
|  | Övriga gyutb. | 305 | 417 | 6 | 728 |  |
|  | Ej klassade gyutb. | 343 | 643 | 13 | 999 |  |
| Total |  | 158452 | 150132 | 557 | 309141 |  |

Tabellen talar om hur många pojkar respektive flickor det finns på ett visst gymnasieprogram. Vi kan med blotta ögat se att antalet pojkar och flickor varierar kraftig på vissa program, exempelvis finns det 11 070 pojkar på Fordonsprogrammet (FP) men endast 313 flickor. För att göra det mer lättöverskådligt kan vi istället för antal utgå från andelar. Vi kan då välja på att antingen beräkna andelarna med kön eller utbildning som bas. Gör samma procedur som tidigare, *Analyze/Descriptive Statistics/Crosstabs...* och som rader (*rows*) välj p2\_3 och som kolumner (*columns*) variabeln kon. Tryck därefter på *Cells*. Tag bort markeringen för *Observed*. Väljer vi att använda kön som bas för procentberäkningen kryssar vi för *Row*. Tryck *Continue* och *Paste*.



Följande skript erhålls:

CROSSTABS

/TABLES=**p2\_3** BY **kon**

/FORMAT= AVALUE TABLES

/CELLS= ROW .

Skillnaden mot ovan är att vi nu specificerar /CELLS till ROW. Tabellen blir nu:

(p2\_3) program 98 \* (kon) kön Crosstabulation

% within (p2\_3) program 98

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | (kon) kön |  |  | Total |  |
|  |  | pojke | Flicka | kön, ej uppgift |  |  |
| (p2\_3) program 98 | BF | 25,4% | 74,5% | ,1% | 100,0% |  |
|  | BP | 97,7% | 2,2% | ,1% | 100,0% |  |
|  | EC | 98,2% | 1,8% | ,0% | 100,0% |  |
|  | EN | 97,4% | 2,5% | ,1% | 100,0% |  |
|  | ES | 32,4% | 67,4% | ,2% | 100,0% |  |
|  | FP | 97,2% | 2,7% | ,0% | 100,0% |  |
|  | HP | 49,8% | 50,1% | ,1% | 100,0% |  |
|  | HV | 14,7% | 85,1% | ,2% | 100,0% |  |
|  | HR | 45,0% | 54,9% | ,0% | 100,0% |  |
|  | IP | 93,9% | 6,1% | ,0% | 100,0% |  |
|  | LP | 40,3% | 59,6% | ,2% | 100,0% |  |
|  | MP | 47,9% | 52,0% | ,1% | 100,0% |  |
|  | NP | 44,3% | 55,5% | ,2% | 100,0% |  |
|  | NV | 59,1% | 40,8% | ,1% | 100,0% |  |
|  | OP | 14,4% | 85,5% | ,1% | 100,0% |  |
|  | SP | 37,6% | 62,2% | ,2% | 100,0% |  |
|  | IB | 35,7% | 62,2% | 2,1% | 100,0% |  |
|  | IV | 56,3% | 43,4% | ,4% | 100,0% |  |
|  | IVIK | 47,4% | 46,5% | 6,1% | 100,0% |  |
|  | SM | 59,9% | 39,8% | ,3% | 100,0% |  |
|  | Övriga gyutb. | 41,9% | 57,3% | ,8% | 100,0% |  |
|  | Ej klassade gyutb. | 34,3% | 64,4% | 1,3% | 100,0% |  |
| Total |  | 51,3% | 48,6% | ,2% | 100,0% |  |

Vi ser nu exempelvis att pojkarna utgör 97,2 % av eleverna på Fordonsprogrammet. Vill vi däremot veta hur många av alla pojkar som går på Fordonsprogrammet väljer vi att visa kolumnprocenten. Vi kan ändra i vårt skript till

CROSSTABS

/TABLES=**p2\_3** BY **kon**

/FORMAT= AVALUE TABLES

/CELLS= COLUMN .

och får då följande tabell:

(p2\_3) program 98 \* (kon) kön Crosstabulation

% within (kon) kön

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | (kon) kön |  |  | Total |  |
|  |  | pojke | flicka | kön, ej uppgift |  |  |
| (p2\_3) program 98 | BF | 2,5% | 7,8% | 3,8% | 5,1% |  |
|  | BP | 3,6% | ,1% | ,5% | 1,9% |  |
|  | EC | 8,1% | ,2% | ,7% | 4,3% |  |
|  | EN | 1,5% | ,0% | ,4% | ,8% |  |
|  | ES | 3,1% | 6,8% | 4,5% | 4,9% |  |
|  | FP | 7,0% | ,2% | ,9% | 3,7% |  |
|  | HP | 4,5% | 4,7% | 2,5% | 4,6% |  |
|  | HV | ,4% | 2,3% | 1,4% | 1,3% |  |
|  | HR | 4,1% | 5,2% | ,9% | 4,6% |  |
|  | IP | 4,1% | ,3% | ,5% | 2,3% |  |
|  | LP | ,5% | ,8% | ,5% | ,6% |  |
|  | MP | 3,3% | 3,8% | 2,2% | 3,5% |  |
|  | NP | 1,9% | 2,6% | 2,2% | 2,2% |  |
|  | NV | 23,3% | 17,0% | 12,2% | 20,2% |  |
|  | OP | ,9% | 5,7% | 1,6% | 3,2% |  |
|  | SP | 18,4% | 32,1% | 23,5% | 25,1% |  |
|  | IB | ,2% | ,4% | 3,8% | ,3% |  |
|  | IV | 5,8% | 4,7% | 11,3% | 5,3% |  |
|  | IVIK | ,4% | ,4% | 15,3% | ,5% |  |
|  | SM | 5,9% | 4,2% | 7,9% | 5,1% |  |
|  | Övriga gyutb. | ,2% | ,3% | 1,1% | ,2% |  |
|  | Ej klassade gyutb. | ,2% | ,4% | 2,3% | ,3% |  |
| Total |  | 100,0% | 100,0% | 100,0% | 100,0% |  |

Vi ser nu att 7,0 % av samtliga pojkar går på Fordonsprogrammet.

Ifall vi vill ha fler uppgifter i samma tabell kan vi kryssa för exempelvis *observed*, *row* och *column* och får då följande skript:

CROSSTABS

/TABLES=**p2\_3** BY **kon**

/FORMAT= AVALUE TABLES

/CELLS= COUNT ROW COLUMN .

Vi kan också köra flera korstabeller på samma gång. I rutan för *rows* kan vi exempelvis sätta in variabeln s2\_1(skollän)och i rutan för *columns* h90soc3(social klass) och får då följande:

CROSSTABS

/TABLES=**p2\_3 s2\_1**  BY **kon h90soc3**

/FORMAT= AVALUE TABLES

/CELLS= COLUMN .

Detta genererar fyra tabeller, 2 x 2. Låt oss se närmare på en av dem, program och social klass. Vi ser längst till höger att av samtliga program är SP-programmet det största, ca 25 % av eleverna går där. Studerar vi sedan fördelningen inom varje klass finner vi att för högre klass är inte SP-programmet det största. NV-programmet har här hela 38 % mot 20 % totalt sett. Samtidigt går endast 11 % av eleverna från Lägre arbetarklass på NV-programmet. En slutsats vi kan dra vad gäller de studieförberedande programmen är att SP har en mer representativ social rekrytering, skillnaderna mellan de sociala klasserna är inte så enormt stora (16 till 32 %), medan NV-programmet är betydligt mer socialt differentierat (11 till 38 %). En omvänd sociala rekryteringen gäller för IV-programmet. Här går totalt 5 % av eleverna, men de som inte finns med bland elever vilkas föräldrar inte finns med i FoB-90 är andelen hela 15 %, samt för högre klass endast 1,5 %. Bland de yrkesförberedande programmen kan vi se att vissa är mer socialt differentierade än andra. Barn och fritidsprogrammet och Fordonsprogrammet har en överrepresentation av arbetarklass och underrepresentation av högre klass. Estetiska programmet och Mediaprogrammet har en betydligt mer jämn social rekrytering.

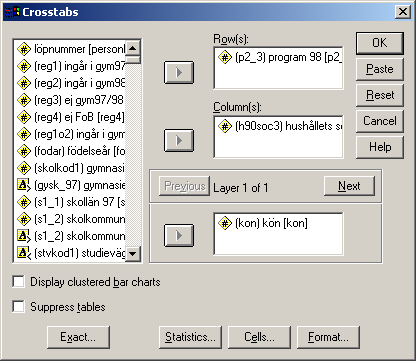
(p2\_3) program 98 \* (h90soc3) hushållets sociala grupp, 5-nivå Crosstabulation

% within (h90soc3) hushållets sociala grupp, 5-nivå

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | (h90soc3) hushållets sociala grupp, 5-nivå | | | |  |  |  |  | Total |  |
|  |  | H-Högre klass | H-Medel-klass | H-Lägre medelklass | H-Högre arbetarklass | H-Lägre arbetarklass | H-Övriga | H-Ej förvärvsarb. | H-Soc-Ej i FoB-90 |  |  |
| (p2\_3) program 98 | BF | 1,8% | 4,0% | 5,3% | 7,2% | 8,2% | 5,3% | 5,7% | 2,7% | 5,1% |  |
|  | BP | ,5% | 1,4% | 2,2% | 3,0% | 2,9% | 2,1% | 2,4% | 1,0% | 1,9% |  |
|  | EC | 2,1% | 3,9% | 4,7% | 6,0% | 5,1% | 3,5% | 4,1% | 3,9% | 4,3% |  |
|  | EN | ,4% | ,7% | ,9% | 1,1% | ,9% | ,9% | ,7% | ,4% | ,8% |  |
|  | ES | 5,1% | 6,0% | 4,6% | 4,3% | 4,5% | 5,6% | 4,9% | 2,0% | 4,9% |  |
|  | FP | ,8% | 2,2% | 4,2% | 5,0% | 6,1% | 3,9% | 4,8% | 5,4% | 3,7% |  |
|  | HP | 2,0% | 3,4% | 5,2% | 5,2% | 5,9% | 4,7% | 6,4% | 8,5% | 4,6% |  |
|  | HV | ,5% | 1,0% | 1,5% | 1,6% | 1,7% | 1,3% | 2,0% | 1,6% | 1,3% |  |
|  | HR | 2,2% | 4,3% | 5,6% | 5,8% | 6,1% | 5,0% | 5,0% | 2,0% | 4,6% |  |
|  | IP | ,7% | 1,5% | 2,3% | 3,6% | 3,7% | 2,0% | 2,3% | 2,0% | 2,3% |  |
|  | LP | ,2% | ,4% | ,7% | ,9% | 1,1% | ,7% | ,8% | ,3% | ,6% |  |
|  | MP | 2,8% | 3,8% | 3,8% | 3,8% | 3,7% | 4,1% | 3,5% | 1,8% | 3,5% |  |
|  | NP | 1,0% | 1,8% | 3,5% | 2,8% | 3,0% | 2,6% | 1,9% | ,3% | 2,2% |  |
|  | NV | 38,1% | 24,4% | 16,9% | 13,5% | 10,8% | 16,8% | 13,7% | 15,0% | 20,2% |  |
|  | OP | 1,1% | 2,3% | 2,8% | 3,9% | 4,5% | 3,0% | 4,8% | 7,9% | 3,2% |  |
|  | SP | 32,0% | 29,3% | 26,0% | 21,6% | 19,0% | 23,8% | 19,5% | 16,3% | 25,1% |  |
|  | IB | ,7% | ,3% | ,2% | ,1% | ,1% | ,5% | ,5% | ,9% | ,3% |  |
|  | IV | 1,5% | 2,9% | 4,0% | 5,0% | 7,9% | 8,5% | 11,8% | 15,3% | 5,3% |  |
|  | IVIK | ,0% | ,0% | ,0% | ,0% | ,0% | ,1% | ,2% | 8,0% | ,5% |  |
|  | SM | 5,7% | 5,7% | 5,2% | 5,1% | 4,3% | 4,7% | 4,2% | 3,6% | 5,1% |  |
|  | Övriga | ,4% | ,3% | ,1% | ,1% | ,1% | ,6% | ,2% | ,5% | ,2% |  |
|  | Ej klassade | ,5% | ,4% | ,3% | ,1% | ,2% | ,3% | ,4% | ,5% | ,3% |  |
| Total |  | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% |  |

### Korstabell med fler än två variabler

Ytterligare en väsentlig användning av korstabellerna är att analysera flera än två variabler samtidigt. Vi har ovan konstaterat att pojkar och flickor fördelar sig mycket olika på olika gymnasieutbildningar. Vi har också sätt att detta gäller för elever med olika socialt ursprung. För att gå vidare i denna analys kan man ställa sig frågan om det är skillnad mellan könen *inom* olika sociala klasser vad gäller vilka program man väljer. För att kunna analysera detta väljer vi att göra kön till en konstanthållen variabel, dvs. vi lägger den som ett lager (*layer*) ovanpå den ursprungliga analysen.



Följande skript erhålls:

CROSSTABS

/TABLES=**p2\_3** BY **h90soc3** BY **kon**

/FORMAT= AVALUE TABLES

/CELLS= COLUMN .

och vi får då följande tabell (något redigerad):(p2\_3) program 98 \* (h90soc3) hushållets sociala grupp, 5-nivå \* (kon) kön Crosstabulation

% within (h90soc3) hushållets sociala grupp, 5-nivå

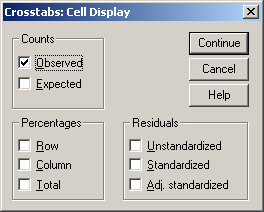
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | (h90soc3) hushållets sociala grupp, 5-nivå | | | |  |  |  |  | Total |  |
| (kon) kön |  |  | H-Högre klass | H-Medel-klass | H-Lägre medelklass | H-Högre arbetarklass | H-Lägre arbetarklass | H-Övriga | H-Ej förvärvsarb. | H-Soc-Ej i FoB-90 |  |  |
| pojke | (p2\_3) | BF | 1,0% | 2,2% | 2,5% | 3,3% | 3,9% | 2,5% | 2,8% | 1,4% | 2,5% |  |
|  |  | BP | 1,0% | 2,6% | 4,2% | 5,6% | 5,5% | 4,1% | 4,6% | 2,0% | 3,6% |  |
|  |  | EC | 3,9% | 7,5% | 9,0% | 11,4% | 9,8% | 6,8% | 8,0% | 7,6% | 8,1% |  |
|  |  | EN | ,7% | 1,3% | 1,8% | 2,1% | 1,8% | 1,5% | 1,3% | ,8% | 1,5% |  |
|  |  | ES | 3,5% | 3,9% | 2,8% | 2,5% | 2,6% | 3,7% | 3,0% | 1,5% | 3,1% |  |
|  |  | FP | 1,5% | 4,1% | 8,1% | 9,4% | 11,5% | 7,4% | 8,9% | 10,5% | 7,0% |  |
|  |  | HP | 2,3% | 3,5% | 5,3% | 4,6% | 5,3% | 4,5% | 6,3% | 8,1% | 4,5% |  |
|  |  | HV | ,1% | ,3% | ,4% | ,5% | ,5% | ,4% | ,6% | ,5% | ,4% |  |
|  |  | HR | 2,3% | 4,0% | 4,6% | 4,7% | 4,9% | 4,5% | 4,3% | 2,5% | 4,1% |  |
|  |  | IP | 1,3% | 2,7% | 4,2% | 6,6% | 7,0% | 3,6% | 4,2% | 3,3% | 4,1% |  |
|  |  | LP | ,1% | ,3% | ,5% | ,8% | ,9% | ,5% | ,7% | ,4% | ,5% |  |
|  |  | MP | 2,8% | 3,7% | 3,4% | 3,3% | 3,3% | 3,9% | 3,5% | 1,8% | 3,3% |  |
|  |  | NP | ,7% | 1,4% | 3,7% | 2,4% | 2,5% | 1,9% | 1,5% | ,2% | 1,9% |  |
|  |  | NV | 43,8% | 28,6% | 19,4% | 15,6% | 12,4% | 19,7% | 15,7% | 15,9% | 23,3% |  |
|  |  | OP | ,3% | ,8% | ,7% | ,9% | 1,3% | ,8% | 1,2% | 2,3% | ,9% |  |
|  |  | SP | 25,4% | 22,3% | 18,4% | 14,2% | 12,9% | 18,8% | 15,2% | 11,0% | 18,4% |  |
|  |  | IB | ,5% | ,2% | ,1% | ,1% | ,1% | ,2% | ,4% | ,7% | ,2% |  |
|  |  | IV | 1,7% | 3,3% | 4,5% | 5,6% | 8,5% | 8,8% | 12,6% | 17,5% | 5,8% |  |
|  |  | IVIK | ,0% | ,0% | ,0% | ,0% | ,1% | ,1% | ,1% | 7,3% | ,4% |  |
|  |  | SM | 6,3% | 6,6% | 6,2% | 6,3% | 5,2% | 5,1% | 4,6% | 4,1% | 5,9% |  |
|  |  | Övriga | ,3% | ,2% | ,1% | ,1% | ,0% | ,6% | ,2% | ,4% | ,2% |  |
|  |  | Ej klassade | ,4% | ,2% | ,2% | ,1% | ,2% | ,2% | ,3% | ,3% | ,2% |  |
|  | Total |  | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% |  |
| flicka | (p2\_3) | BF | 2,7% | 5,8% | 8,3% | 11,4% | 12,7% | 8,1% | 8,7% | 4,0% | 7,8% |  |
|  |  | BP | ,0% | ,1% | ,1% | ,1% | ,2% | ,1% | ,1% | ,0% | ,1% |  |
|  |  | EC | ,1% | ,2% | ,1% | ,2% | ,2% | ,2% | ,2% | ,1% | ,2% |  |
|  |  | EN | ,0% | ,0% | ,0% | ,0% | ,0% | ,2% | ,1% |  | ,0% |  |
|  |  | ES | 6,7% | 8,2% | 6,6% | 6,3% | 6,5% | 7,5% | 6,9% | 2,4% | 6,8% |  |
|  |  | FP | ,0% | ,1% | ,2% | ,3% | ,5% | ,3% | ,4% | ,0% | ,2% |  |
|  |  | HP | 1,6% | 3,3% | 5,2% | 5,9% | 6,5% | 4,9% | 6,5% | 9,3% | 4,7% |  |
|  |  | HV | ,9% | 1,7% | 2,6% | 2,9% | 3,0% | 2,1% | 3,4% | 2,9% | 2,3% |  |
|  |  | HR | 2,1% | 4,5% | 6,6% | 6,9% | 7,3% | 5,5% | 5,7% | 1,6% | 5,2% |  |
|  |  | IP | ,1% | ,2% | ,2% | ,4% | ,4% | ,4% | ,4% | ,7% | ,3% |  |
|  |  | LP | ,2% | ,5% | ,9% | 1,1% | 1,4% | ,8% | 1,0% | ,3% | ,8% |  |
|  |  | MP | 2,8% | 4,0% | 4,1% | 4,3% | 4,1% | 4,2% | 3,5% | 1,7% | 3,8% |  |
|  |  | NP | 1,3% | 2,2% | 3,4% | 3,2% | 3,5% | 3,3% | 2,4% | ,3% | 2,6% |  |
|  |  | NV | 32,2% | 20,0% | 14,3% | 11,3% | 9,2% | 13,9% | 11,7% | 14,2% | 17,0% |  |
|  |  | OP | 1,9% | 3,8% | 5,1% | 7,1% | 7,8% | 5,3% | 8,5% | 14,6% | 5,7% |  |
|  |  | SP | 39,1% | 36,8% | 34,0% | 29,6% | 25,4% | 28,9% | 24,1% | 21,5% | 32,1% |  |
|  |  | IB | ,9% | ,4% | ,3% | ,1% | ,1% | ,8% | ,6% | 1,0% | ,4% |  |
|  |  | IV | 1,2% | 2,6% | 3,5% | 4,5% | 7,3% | 8,2% | 11,0% | 13,3% | 4,7% |  |
|  |  | IVIK | ,0% | ,0% | ,0% | ,0% | ,0% | ,1% | ,2% | 8,1% | ,4% |  |
|  |  | SM | 5,1% | 4,7% | 4,1% | 3,9% | 3,3% | 4,2% | 3,7% | 2,8% | 4,2% |  |
|  |  | Övriga | ,4% | ,3% | ,2% | ,1% | ,1% | ,6% | ,3% | ,6% | ,3% |  |
|  |  | Ej klassade | ,6% | ,6% | ,4% | ,2% | ,3% | ,5% | ,5% | ,5% | ,4% |  |
|  | Total |  | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% |  |

Vi kan här jämföra pojkars och flickors programval *inom* samma klass. Vi såg ovan att pojkar totalt sett i större utsträckning än flickor väljer NV-programmet och att högre klass i större utsträckning väljer NV-programmet än andra. När vi nu beaktar både kön och social klass ser vi – inte förvånade – att pojkar ur den högre klassen är de som i störst utsträckning prioriterar NV. Men vad vi vidare ser är att skillnaderna mellan könen i benägenhet att gå på NV-programmet faktiskt är som störst i den högre klassen följda av medelklassen, hela 11 resp. 9 procentenheter skiljer pojkarna från flickorna (däremot skiljer sig förhållandet mellan pojkarnas andel och flickornas andel så mycket mellan klasserna, kvoten är ungefär 1,5 för samtliga). Noterbart är också den stora skillnaden som råder mellan könen i den lägre arbetarklassen. Pojkarna fördelar sig relativt jämt på NV-programmet och SP-programmet (12,4 % och 12,9 %) medan för flickorna är SP-programmet det klart dominerande valet (25,4 % mot 9,2 %).

Man kan vidare även lägga på ytterligare lager i analysen. Välj då Next Layer och klicka in den variabel som önskas. Problemet är dock att det blir rätt komplicerat att analysera resultaten ju fler lager man lägger på. Ofta kan det vara lämpligt att hålla sig till korstabeller med två variabler och ett lager och prova ut olika varianter av analysen. Man kan på så sätt exempelvis analysera förhållande mellan sociologiska grundvariabler som ålder, kön, klass, och betyg, program- och skolval.

### Något om olika inställningar och statistiska mått i samband med korstabeller

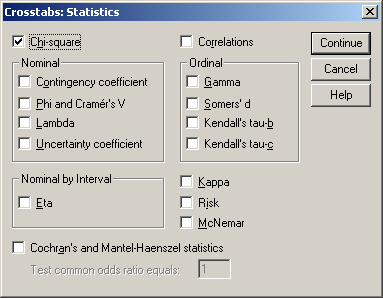
När man går in och specificerar sin korstabell i menyn finns en rad ytterligare mått och inställningar att använda. I rutan *Cells* kan man exempelvis välja att visa *Expected Value*, dvs. det värde cellen skulle ha om alla värden var statistiskt sett oberoende (formel). Genom att göra två tabeller, en med faktiskt värde, *Observed*, och en med förväntat värde, *Expected*, kan man sedan i Excel räkna ut hur pass över- eller underrepresenterad en viss grupp är (formel). Det finns även en variant av detta i SPSS, genom att välja residuals får man olika mått på hur pass över- eller underrepresenterade en grupp är (formler).



Man kan även få ett antal statistiska mått för sin korstabellen. Genom att gå in på *Statistics* kan man specificera de mått man är intresserad av. Här är det viktigt att veta vad det är för typ av variabler man använder.

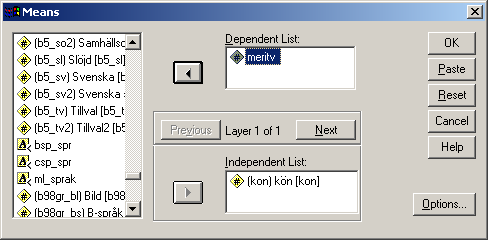
* Nominala variabler är variabler som delar in olika egenskaper som inte kan klassificeras hierarkiskt eller rangordnas. Födelselän eller kön är vanligtvis nominala variabler.
* Ordinala variabler kan klassificeras hierarkiskt, betyg av typen MVG, VG och G är ordinala data.
* Intervallvariabler har jämna intervall mellan varje steg, temperaturen mätt i Celsius är ett exempel där avståndet mellan varje grad är lika.
* Kvotvariabler har också jämna intervall mellan värden men dessutom en 0-punkt.[[7]](#footnote-7)

Vanligtvis arbetar man med nominal och ordinaldata. De vanligaste måtten för att beräkna samband mellan två variabler är för nominaldata Lambda (*Lambda*) och för ordinaldata Kendalls’s T (*Kendall’s tau b*) och Spearmans korrelation (*Correlations*). *Chi-2* kan med fördel användas för både nominal och ordinaldata.[[8]](#footnote-8)



# Jämförelse av medelvärden

När man har variabler som ålder och inkomst är man ibland intresserad av att jämföra olika gruppers medelvärden. För att göra detta går man in på *Analyze/Compare Means/Means*. I rutan för *Dependent List* väljs de variabler som man vill beräkna medelvärdet av. I rutan för *Imdependent List* väljs de variabler som man vill jämföra. Vi har i exemplet nedan valt elevernas meritvärde (avgångsbetyg från gymnasiet år 1998) och kön.



Skriptet för denna analys ser ut som följer:

MEANS

TABLES=**meritv** BY **kon**

/CELLS MEAN COUNT STDDEV .

Vi har här specificerat att vi vill ha medelvärdet, antalet individer och standardavvikelsen. Genom att gå in på *Options* kan man även välja exempelvis medianvärdet, variansen, högsta respektive lägsta värde. Den tabell vi erhåller i vårt exempel visas nedan:

**Means**

****

****

Först ser vi att endast 22,8 % av hela populationen har uppgifter om meritvärde och kön. Av de som vi har uppgifter om finns en tydlig skillnad mellan könen. Flickor får 31 poäng högre medelvärde än pojkar (totalt kan man ha ett meritvärde på 320). Standardavvikelsen visar att det inte är någon större skillnad mellan könen vad gäller spridningen av resultaten.

Även här kan vi analysera olika lager av variabler. Vi kan exempelvis lägga in social klass för att analysera förhållandet mellan kön, meritvärde och social klass. Vi kan då exempelvis skriva (det går bra att byta plats på h90soc3 och kon):

MEANS

TABLES=**meritv** BY **h90soc3** BY **kon**

/CELLS MEAN COUNT STDDEV .

Den tabell vi erhåller visar dels att inom varje social klass har flickorna högre meritvärde än pojkarna, dels att ju högre social klass desto högre meritvärde har eleverna.

Report

MERITV

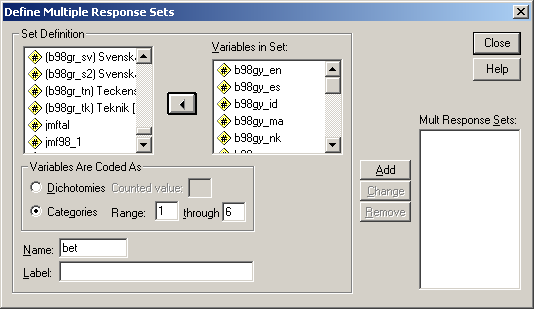
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| (h90soc3) hushållets sociala grupp, 5-nivå | (kon) kön | Mean | N | Std. Deviation |  |
| H-Högre klass | pojke | 220,66 | 7301 | 51,749 |  |
|  | flicka | 239,65 | 6805 | 50,471 |  |
|  | Total | 229,82 | 14106 | 52,007 |  |
| H-Medelklass | pojke | 200,88 | 12853 | 51,925 |  |
|  | flicka | 222,13 | 12237 | 53,403 |  |
|  | Total | 211,25 | 25090 | 53,711 |  |
| H-Lägre medelklass | pojke | 188,51 | 7552 | 52,316 |  |
|  | flicka | 210,49 | 7025 | 52,946 |  |
|  | Total | 199,10 | 14577 | 53,753 |  |
| H-Högre arbetarklass | pojke | 180,46 | 7125 | 52,212 |  |
|  | flicka | 203,63 | 6778 | 52,859 |  |
|  | Total | 191,75 | 13903 | 53,788 |  |
| H-Lägre arbetarklass | pojke | 169,87 | 8639 | 55,892 |  |
|  | flicka | 191,97 | 8485 | 58,218 |  |
|  | Total | 180,82 | 17124 | 58,115 |  |
| H-Övriga | pojke | 172,84 | 1736 | 63,154 |  |
|  | flicka | 195,51 | 1653 | 63,865 |  |
|  | Total | 183,90 | 3389 | 64,496 |  |
| H-Ej förvärvsarb. | pojke | 163,39 | 2276 | 64,842 |  |
|  | flicka | 181,81 | 2227 | 67,057 |  |
|  | Total | 172,50 | 4503 | 66,579 |  |
| H-Soc-Ej i FoB-90 | pojke | 154,02 | 2165 | 72,322 |  |
|  | flicka | 171,09 | 1986 | 75,085 |  |
|  | Total | 162,19 | 4151 | 74,140 |  |
| Total | pojke | 188,84 | 49647 | 57,909 |  |
|  | flicka | 209,86 | 47196 | 58,856 |  |
|  | Total | 199,08 | 96843 | 59,311 |  |

# Multipla svar-tabeller

Ibland har variabler som inte är logiskt uteslutande och klart strukturerade. Vi kan ta högskolestudier över ett antal terminer som exempel. Det finns hundratals olika utbildning i högskolan och varje individ kan ha gått på flera olika utbildningar även samma termin. För att få reda på hur många totalt som läst på en viss utbildning mellan åren 1993 och 1998 kan vi skapa göra en analys av multipla svar.

OBS att det måste vara numeriska variabler och att samma värde gäller för alla variabler, exempelvis att gå på en ekonomiutbildning måste ha samma värde i alla variabler. Ifall du vill analysera strängvariabler kan du enkelt omkoda dessa till numeriska värden genom *Automatic Recode*, se Avsnitt IX.2.8 Automatisk omkodning av strängvariabel till numerisk variabel.

Här ska vi analysera olika betygsvärden. Dessa är samma för alla betyg, men individerna har fått olika antal betyg, och vi är här intresserade av hur många som har fått MVG i relation till alla utgivna betyg. Vi går då in på *Analyze/Multiple Responses/Define Set...* Först definierar vi variablerna i setet (samtliga variabler som börjar med b98gy). Därefter de värden vi vill räkna (1 till 6). Sedan döper vi det nya setet av variabler (här bet) och trycker på *Add*.



Nästa steg är att gå in på *Analyze/Multiple Responses/Frequensis...* Vi väljer där vårt set av variabler bet. Följande skript erhålls:

MULT RESPONSE

GROUPS**=$bet** (**b98gy\_en b98gy\_es b98gy\_id b98gy\_ma b98gy\_nk b98gy\_re b98gy\_sh**

**b98gy\_sv b98gy\_sl b98gy\_ss b98gy\_ty b98gy\_fr b98gy\_tc b98gy\_fc** (1,6))

/FREQUENCIES=**$bet**  .

Om vi kör skriptet får vi följande tabell:

**Multiple Response**

Group $BET

Pct of Pct of

Category label Code Count Responses Cases

en-ej läst 1 280350 25,8 360,8

en-\* 2 409 ,0 ,5

en-IG 3 33130 3,0 42,6

en-G 4 380080 34,9 489,2

en-VG 5 296405 27,2 381,5

en-MVG 6 97440 9,0 125,4

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

Total responses 1087814 100,0 1400,0

347.713 missing cases; 77.701 valid cases

Vi ser här att av alla betyg som delats ut utgör MVG 9,0 %. G är det klart vanligaste betyget. Tabellen är dock något missvisande eftersom de som inte har läst ett visst ämne finns med. Vi kan korrigera detta i Excel och räkna ut andelen MVG på alla betyg exklusive de som inte har läst ett visst ämne. Andelen MVG stiger nu till 12,1 %.

Vi kan även göra en korstabell genom att gå in på *Analyze/Multiple Responses/Crosstabs...* I Options kan man specificera att man vill ha rad-, kolumn- och totalprocent, baserade på *Responses* (man kan även välja *Cases*). Skriptet blir då:

MULT RESPONSE

GROUPS=**$bet** (**b98gy\_en b98gy\_es b98gy\_id b98gy\_ma b98gy\_nk b98gy\_re b98gy\_sh**

**b98gy\_sv b98gy\_sl b98gy\_ss b98gy\_ty b98gy\_fr b98gy\_tc b98gy\_fc** (1,6))

/VARIABLES=**kon**(1 2)

/TABLES=**$bet** BY **kon**

/CELLS=ROW COLUMN TOTAL

/BASE=RESPONSES .

Vi ser i tabellen nedan att flickorna har 60,7 % av alla MVG och att av alla flickornas betyg utgör MVG 10,7 %. Motsvarande siffror för pojkarna är 39,3 % och 7,1 %. Även här bör vi dock korrigera för dem som inte läst ett visst ämne.

**Multiple Response**

\* \* \* C R O S S T A B U L A T I O N \* \* \*

$BET (group)

by KON (kon) kön

KON

Count pojke flicka

Row pct  Row

Col pct  Total

Tab pct  1  2 

$BET 

1 141694 138584 280278

en-ej läst  50,6  49,4  25,8

 26,5  25,1 

 13,0  12,7 



2  279  126  405

en-\*  68,9  31,1  ,0

 ,1  ,0 

 ,0  ,0 



3  21727  11397  33124

en-IG  65,6  34,4  3,0

 4,1  2,1 

 2,0  1,0 



4 207991 172027 380018

en-G  54,7  45,3  34,9

 38,8  31,2 

 19,1  15,8 



5 125609 170745 296354

en-VG  42,4  57,6  27,2

 23,5  30,9 

 11,5  15,7 



6  38256  59169  97425

en-MVG  39,3  60,7  9,0

 7,1  10,7 

 3,5  5,4 



Column 535556 552048 1087604

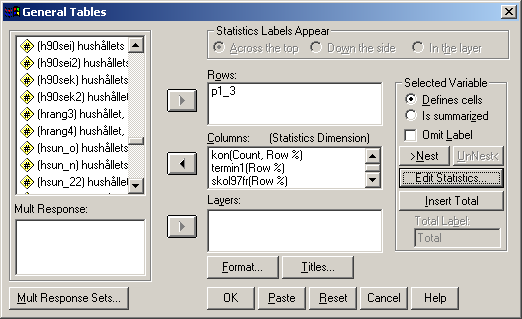
Total 49,2 50,8 100,0

Percents and totals based on responses

77.686 valid cases; 347.728 missing cases

# Generella tabeller

När vi vill skapa en översikt över exempel ett antal utbildningar kan vi med fördel utnyttja funktionen *General Tables*, som nås via *Analyze/Custom Tabels/ General Tables...* Vi börjar med att specificera vilken variabel vi vill ha på raderna. I det här fallet väljer vi p1\_3. Därefter väljer vi ett antal variabler som vi vill korstabulera, här kon + termin1 + skol97fr + mbt + h90soc3. Vi kan sedan gå in på *Edit Statistics* och välja vilken information vi vill ha. Här har jag valt radprocenten.



Vi får följande skript:

\* General Tables.

TABLES

/FORMAT BLANK MISSING('.')

/GBASE=CASES

/TABLE=p1\_3 BY **kon + termin1 + skol97fr + mbt + h90soc3**

/STATISTICS

count( **kon**( F5.0 ))

cpct( **kon**( PCT5.1 ) 'Row %':p1\_3 )

cpct( **termin1**( PCT5.1 ) 'Row %':p1\_3 )

cpct( **skol97fr**( PCT5.1 ) 'Row %':p1\_3 )

cpct( **mbt**( PCT5.1 ) 'Row %':p1\_3 )

cpct( **h90soc3**( PCT5.1 ) 'Row %':p1\_3 ).

Detta ger en tabell som redigerad återges nedan. Vi ser att vi på ett enkelt sett kan samla en rad olika typer av information om en viss utbildning. Genom att klippa in tabellen i Excel kan vi sedan sortera utbildningarna efter vilka värden de uppvisar på de olika variablerna. Det går även lätt att redigera tabellen.

1. Program 1997 och kön, termin, skoltyp och betyg (modifierad)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | (kon) kön | | | | (termin1) termin 97 | | | (skol97fr) fristående skola 97 | | (mbt) grundskola medelbetyg | | | |
|  |  | pojke |  | flicka |  | 1 | 3 | 5 | Frist. gy.sk. | Kom. gy.sk. | Ej m-bet i gr.sk. | M.bet gr <2,5 | M.bet gr 2,5-2,9 | M.bet gr 4,5-5,0 |
|  |  | Count | Row % | Count | Row % | Row % | Row % | Row % | Row % | Row % | Row % | Row % | Row % | Row % |
| (p1\_3) | BF | 4851 | 25,1% | 14424 | 74,8% | 31,6% | 33,7% | 34,6% |  | 100,0% | ,2% | 21,7% | 32,9% | ,0% |
|  | BP | 6747 | 97,3% | 183 | 2,6% | 31,5% | 34,2% | 34,2% | ,4% | 99,6% | ,1% | 40,2% | 31,3% | ,0% |
|  | EC | 13475 | 98,5% | 201 | 1,5% | 36,5% | 32,5% | 31,0% |  | 100,0% | ,1% | 13,5% | 34,3% | ,1% |
|  | EN | 2392 | 97,4% | 63 | 2,6% | 38,9% | 32,3% | 28,8% |  | 100,0% | ,3% | 28,8% | 34,2% |  |
|  | ES | 5033 | 33,0% | 10209 | 66,8% | 37,9% | 32,5% | 29,6% | 2,4% | 97,6% | ,7% | 8,1% | 18,3% | 1,6% |
|  | FP | 11804 | 97,5% | 292 | 2,4% | 37,1% | 32,5% | 30,4% | ,4% | 99,6% | ,3% | 38,0% | 32,4% | ,0% |
|  | HP | 7892 | 48,5% | 8350 | 51,3% | 35,2% | 32,2% | 32,6% | ,6% | 99,4% | ,3% | 21,0% | 35,4% | ,0% |
|  | HV | 613 | 15,1% | 3436 | 84,8% | 36,8% | 33,5% | 29,7% | 1,6% | 98,4% | ,2% | 11,4% | 16,5% | ,2% |
|  | HR | 6323 | 44,9% | 7753 | 55,1% | 38,2% | 32,6% | 29,2% | 1,2% | 98,8% | ,3% | 11,1% | 29,2% | ,0% |
|  | IP | 7568 | 93,5% | 516 | 6,4% | 30,7% | 34,9% | 34,4% | ,4% | 99,6% | ,2% | 42,9% | 28,0% | ,1% |
|  | LP | 871 | 40,4% | 1283 | 59,5% | 37,0% | 33,4% | 29,6% |  | 100,0% | ,2% | 26,5% | 33,1% |  |
|  | MP | 4586 | 45,1% | 5568 | 54,8% | 35,6% | 33,0% | 31,4% | 9,7% | 90,3% | ,3% | 5,0% | 16,0% | ,3% |
|  | NP | 3216 | 45,7% | 3810 | 54,2% | 37,7% | 32,3% | 30,0% |  | 100,0% | ,2% | 21,4% | 26,6% | ,3% |
|  | NV | 36766 | 59,5% | 24913 | 40,3% | 36,0% | 33,8% | 30,2% | 1,9% | 98,1% | ,4% | ,3% | 2,2% | 11,0% |
|  | OP | 1666 | 14,7% | 9627 | 85,1% | 32,6% | 33,2% | 34,3% | 1,3% | 98,7% | ,4% | 20,1% | 28,3% | ,1% |
|  | SP | 29466 | 37,9% | 48052 | 61,9% | 34,0% | 33,3% | 32,7% | 2,8% | 97,2% | ,4% | 1,7% | 9,9% | 1,9% |
|  | IB | 327 | 36,1% | 531 | 58,6% | 38,3% | 38,4% | 23,3% | 18,7% | 81,3% | 3,9% | ,3% | ,6% | 16,6% |
|  | IV | 7868 | 55,0% | 6375 | 44,5% | 84,4% | 10,2% | 5,4% | ,2% | 99,8% | 3,2% | 49,6% | 13,8% | ,1% |
|  | IVIK | 665 | 49,1% | 616 | 45,5% | 96,9% | 3,0% | ,1% |  | 100,0% | 5,8% | 19,6% | 6,4% | ,1% |
|  | SM | 6885 | 59,4% | 4677 | 40,4% | 43,3% | 34,7% | 22,0% | 13,7% | 86,3% | 1,1% | 8,7% | 15,7% | 2,5% |
|  | Övriga | 282 | 40,9% | 384 | 55,7% | 34,0% | 32,7% | 29,3% | 100,0% |  | 73,7% | 1,3% | 1,6% | ,3% |
|  | Ej klassade | 40 | 19,3% | 148 | 71,5% | 38,2% | 32,9% | 29,0% | 100,0% |  |  | ,5% | ,5% |  |

# Omkodning, beräkningar, etc. av variabler

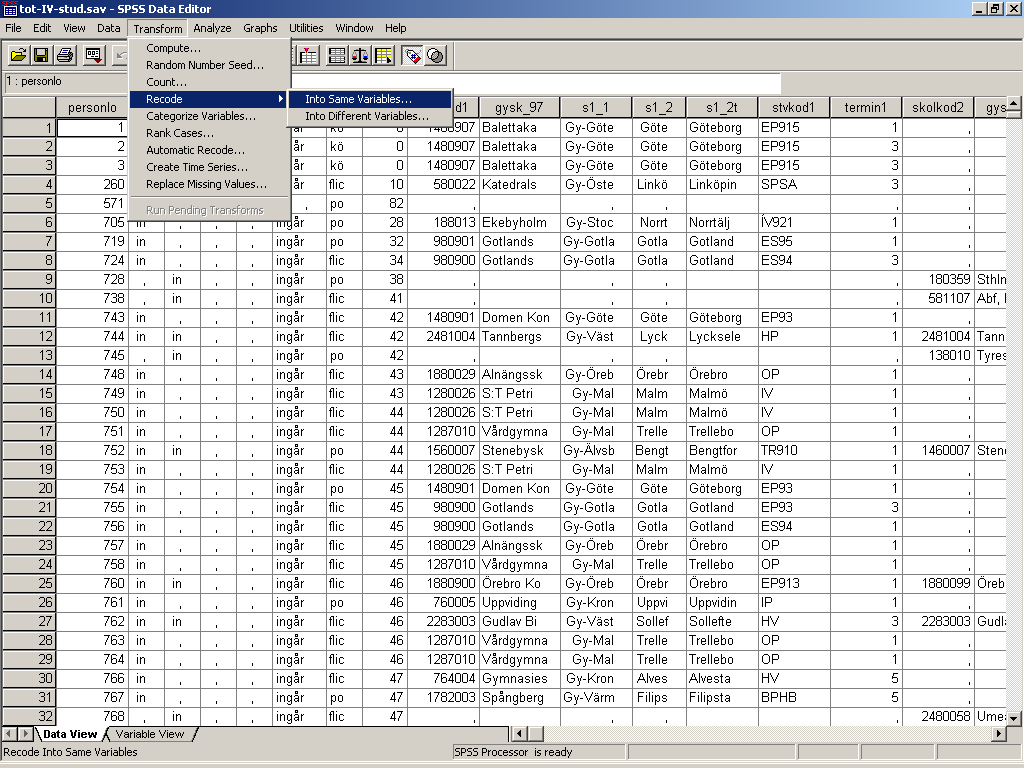
Det krävs nästintill alltid att man måste koda om variabler för att kunna göra de analyser man önskar. Detta går att göra på flera sätt. För den som inte arbetar så mycket med SPSS och har små dataset går det bra att använda menystyrd programmering. För övriga rekommenderar jag att man använder skript för att göra omkodningar. Framställningen kommer att koncentreras på det senare sättet, men låt oss börja med hur man går till väga via menyn.

## Omkodning av samma variabel...

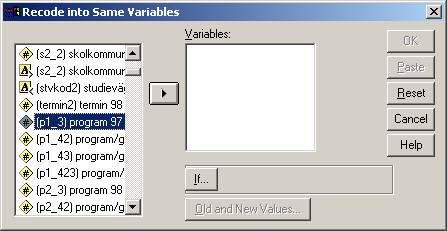
Ibland vill man ändra värdena i en variabel, exempelvis när man lagt in värden manuellt och gett samma egenskap olika värden (dvs. egenskapen ”äga en Volvo” är kodad både som värde 2 och 7). Ibland vill man slå samman olika kategorier såsom ”ej svar” och ”vet ej”, men här rekommenderar jag att man istället skapar en ny variabel. Nackdelen med att koda om i samma variabel är att de ursprungliga värdena försvinner (den ursprungliga skillnaden mellan ”ej svar” och ”vet ej” kanske man inser efteråt att man vill granska närmare, vilket inte går om man slagit samman dessa). Hur går man då till väga för att koda om en variabel?

### ... via menyn

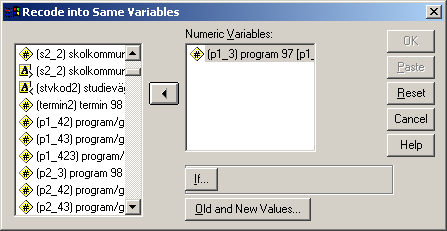
Via menyn går man in på *Transform/Recode/Into Same Variables...* .



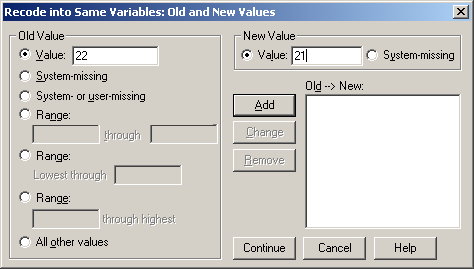
Därefter kommer följande fönster upp:



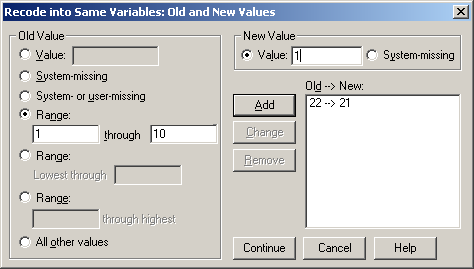
Välj nu en variabel att koda om, här väljer jag p1\_3 genom att först leta upp och markera den variabel och sedan trycka på pilsymbolen.



Nästa steg är att bestämma hur variabeln skall kodas om. Detta görs genom att trycka på knappen *Old and New Values*.



Jag börjar med att specificera det värde jag vill koda om (*Old value*), i detta fall värdet 22. Därefter specificerar jag det nya värdet (*New value*): 21. Sedan trycker jag på *Add*. Nu kan jag välja ett nytt värde att koda om. Observera att det går att koda om flera värden åt gången. Genom att exempelvis välja *Range* kan jag samtidigt koda om flera variabler (förutsatt att de alla ska få ett enda värde):



Man kan även välja att koda om spannet det lägsta värdet – ett visst värde (*Range Lowest through \_\_\_*), eller ett visst värde – det högsta värdet (*Range \_\_\_ through Highest*), övriga värden (All other values), samt bortfall (*System-missing*, *System-missing or user-missing*).

När man sedan valt hur man ska koda om sina värden klickar man på *Continue*. Därefter klicka på *OK* så kodas variabeln om som man specificerat det (övriga värden förbli desamma). Observera att för att slå samman två värden, låt säga 1 och 2 räcker det med att koda om värdet 2 till 1 (eftersom värdet 1 naturligtvis redan har värdet 1).

### ... via skript

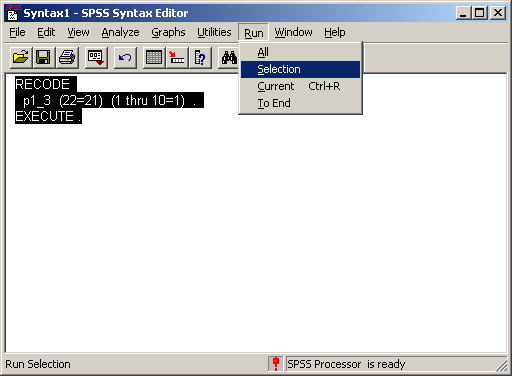
Vanligen är det bättre att koda om variabler via skript. I exemplet ovan kan vi när vi har specificerat de olika värdena som skall kodas om välja att trycka på *Paste* (klistra in). De specifikationer vi gjort kommer då att formuleras som ett skript som visas i *Syntax Editor* (syntaxredigeraren) enligt följande:

RECODE

**p1\_3** (22=21) (1 thru 10=1) .

EXECUTE .

För att köra skriptet, markera skriptet och tryck på *Run/Selection*.



Poängen med syntaxredigeraren är att vi inte behöver gå in via menyn och specificera alla inställningar. Detta kan vi istället göra direkt i syntaxredigeraren. Följande kan vara bra att känna till:

För att koda om ett värde till ett annat skriv (**V1**=**V2**), där V1 står för värde 1 och V2 för värde 2.

För att koda om fler värden i rad till ett annat värde skriv (**V1** thru **V2**=**V3**).

För att koda om spannet det lägsta värdet till ett specifikt värde till ett nytt värde skriv (Lowest thru **V1**=**V2**).

För att koda om spannet ett specifikt värde till det högsta värdet till ett nytt värde skriv (**V1** thru Highest =**V2**).

För att koda om *System-missing* till ett visst värde skriv (SYSMIS=**V1**) och för att koda om ett specifikt värde till *System-missing* skriv (**V1**=SYSMIS).

För att koda om *System-missing or user-missing* till ett visst värde skriv (MISSING=**V1**) och för att koda om ett specifikt värde till *System-missing or user-missing* skriv (**V1**=MISSING).

För att koda om övriga värden (när man redan specificerat ett antal värden som ska kodas om) skriv (ELSE=**V1**).

För att koda om ett spann av värden till samma värden skriver man (**V1** thru **V2**=COPY). (Detta är dock inte meningsfullt förrän man kodar om en variabel till en annan (ny) variabel, se avsnitt IX.2 Omkodning, beräkningar, etc. av en variabel till en annan variabel).

(OBS! All text med fet stil måste bytas mot faktiska värden, övriga beteckningar får inte ändras)

Låt oss ta ett fiktivt exempel. Vi vill aggregera en variabel med 12 värden + *System missing* till 6 värden. Vi kan då koda om variabeln på följande sätt:

RECODE

**p1\_3** (1 thru 2=1) (3 thru 5=2) (6=3) (7 thru 8=4) (9 thru highest=5) (SYSMIS=6) .

EXECUTE .

Observera att varje omkodning måste omslutas av en parantes och att efter den sista parantes måste en punkt komma. Därefter måste det vara en radbrytning (enter-slag). Det krävs också en punkt efter EXECUTE.

För att kontrollera resultatet är det bra att dra ut en frekvenstabell (se kapitel IV Frekvenstabeller).

När är det då lämpligt att göra omkodningar av en och samma variabel? Detta kan man göra när man vill:

* korrigera kodningsfel (exempelvis en och samma företeelse förekommer med två olika värden
* anpassa kodningen så att den fungerar för SPAD (se kapitel XVII Att koda data för överföring till SPAD (program för bland annat korrespondensanalys))
* ändra *System missing* till givet värde (eller vice versa)

## Omkodning, beräkningar, etc. av en variabel till en annan variabel

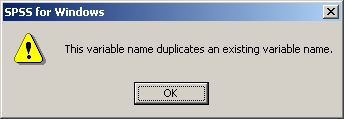
Det finns ett antal tillfällen då det är nödvändigt att koda om en eller flera variabler till en annan (ny) variabel. Här är de viktigaste områdena:

* För att göra mer aggregerade varianter
* För att skapa kombinerade variabler
* För att göra beräkningar av olika slag
* För att göra om en numerisk variabel till en strängvariabel
* För att skapa kombinerade strängvariabler
* För att skapa filtervariabler

De fem första områdena kommer att beaktas i detta avsnitt, det sjätte i kapitel X Filter.

### Placering av ny variabel

När vi skapar en ny variabel hamnar denna per default sist i dataarket. Detta kan vara opraktiskt, ofta vill man ha variabler som hänger samman efter varandra i arket. Detta kan lösas på två sätt. Antingen flyttar man den nya variabeln som har lagts sist i dataarket till den position man vill ha den på, eller så skapar man en ny variabel innan man börjar omkodningen, se kapitel Inledning. Väljer man det senare alternativet döper man den nya variabeln till det namn man ämnar omkoda variabeln till. Använder man menyn för att koda om variabeln kommer det upp en varning när man skriver in namnet på den nya variabeln (den finns ju redan i dataarket):



Tryck då bara på *OK* och fortsätt.

### Aggregation av variabel

När man kodar data eller använder registerdata är fördelningen av variabelvärden ofta dåligt anpassad för vidare analyser. Ett konkret exempel är gymnasieelevernas födelseår. Drar vi ut en frekvens över födelseår erhåller vi följande fördelning:

1. Födelseår, elever i gymnasieskolan 1997—1998.[[9]](#footnote-9)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| (fodar) födelseår | | Frequency | Percent | Valid Percent | Cumulative Percent |
| Valid | 0 | 3 | 0,000705 | 0,000705 | 0,000705 |
|  | 10 | 1 | 0,000235 | 0,000235 | 0,00094 |
|  | 28 | 1 | 0,000235 | 0,000235 | 0,001175 |
|  | 32 | 1 | 0,000235 | 0,000235 | 0,00141 |
|  | 34 | 1 | 0,000235 | 0,000235 | 0,001645 |
|  | 38 | 1 | 0,000235 | 0,000235 | 0,001881 |
|  | 41 | 1 | 0,000235 | 0,000235 | 0,002116 |
|  | 42 | 3 | 0,000705 | 0,000705 | 0,002821 |
|  | 43 | 2 | 0,00047 | 0,00047 | 0,003291 |
|  | 44 | 4 | 0,00094 | 0,00094 | 0,004231 |
|  | 45 | 5 | 0,001175 | 0,001175 | 0,005406 |
|  | 46 | 5 | 0,001175 | 0,001175 | 0,006582 |
|  | 47 | 5 | 0,001175 | 0,001175 | 0,007757 |
|  | 48 | 5 | 0,001175 | 0,001175 | 0,008932 |
|  | 49 | 5 | 0,001175 | 0,001175 | 0,010108 |
|  | 50 | 7 | 0,001645 | 0,001645 | 0,011753 |
|  | 51 | 14 | 0,003291 | 0,003291 | 0,015044 |
|  | 52 | 15 | 0,003526 | 0,003526 | 0,01857 |
|  | 53 | 21 | 0,004936 | 0,004936 | 0,023507 |
|  | 54 | 24 | 0,005642 | 0,005642 | 0,029148 |
|  | 55 | 14 | 0,003291 | 0,003291 | 0,032439 |
|  | 56 | 25 | 0,005877 | 0,005877 | 0,038316 |
|  | 57 | 24 | 0,005642 | 0,005642 | 0,043957 |
|  | 58 | 30 | 0,007052 | 0,007052 | 0,051009 |
|  | 59 | 35 | 0,008227 | 0,008227 | 0,059236 |
|  | 60 | 46 | 0,010813 | 0,010813 | 0,070049 |
|  | 61 | 44 | 0,010343 | 0,010343 | 0,080392 |
|  | 62 | 59 | 0,013869 | 0,013869 | 0,094261 |
|  | 63 | 61 | 0,014339 | 0,014339 | 0,1086 |
|  | 64 | 66 | 0,015514 | 0,015514 | 0,124114 |
|  | 65 | 64 | 0,015044 | 0,015044 | 0,139159 |
|  | 66 | 64 | 0,015044 | 0,015044 | 0,154203 |
|  | 67 | 82 | 0,019275 | 0,019275 | 0,173478 |
|  | 68 | 89 | 0,020921 | 0,020921 | 0,194399 |
|  | 69 | 91 | 0,021391 | 0,021391 | 0,21579 |
|  | 70 | 136 | 0,031969 | 0,031969 | 0,247759 |
|  | 71 | 171 | 0,040196 | 0,040196 | 0,287955 |
|  | 72 | 217 | 0,051009 | 0,051009 | 0,338964 |
|  | 73 | 284 | 0,066758 | 0,066758 | 0,405722 |
|  | 74 | 410 | 0,096377 | 0,096377 | 0,502099 |
|  | 75 | 724 | 0,170187 | 0,170187 | 0,672286 |
|  | 76 | 1607 | 0,37775 | 0,37775 | 1,050036 |
|  | 77 | 4312 | 1,013601 | 1,013601 | 2,063637 |
|  | 78 | 19441 | 4,569901 | 4,569901 | 6,633538 |
|  | 79 | 100211 | 23,55611 | 23,55611 | 30,18965 |
|  | 80 | 102591 | 24,11557 | 24,11557 | 54,30522 |
|  | 81 | 99383 | 23,36148 | 23,36148 | 77,6667 |
|  | 82 | 94169 | 22,13585 | 22,13585 | 99,80255 |
|  | 83 | 815 | 0,191578 | 0,191578 | 99,99412 |
|  | 84 | 15 | 0,003526 | 0,003526 | 99,99765 |
|  | 85 | 4 | 0,00094 | 0,00094 | 99,99859 |
|  | 96 | 3 | 0,000705 | 0,000705 | 99,99929 |
|  | 97 | 2 | 0,00047 | 0,00047 | 99,99976 |
|  | 99 | 1 | 0,000235 | 0,000235 | 100 |
|  | Total | 425414 | 100 | 100 |  |

Vi ser då snabbt att även om födelseåren är många, totalt 54 stycken, är fyra årtal nästintill helt dominerande, 79, 80, 81 och 82. Dessa fyra årtal svara för 23,6, 24,1, 23,4 respektive 22,2 %, eller totalt 93,2 % av alla elever. Det är då knappast meningsfullt att behålla alla 54 årtalskategorierna, utan vi bör koda om dessa på något sätt. Ett förslag är att koda födelsetalen i 7 grupper: 1) alla födda före 76, 2) alla födda mellan 76 och 78, 3) alla födda 79, 4) alla födda 80, 5) alla födda 81, 6) alla födda 82, 7) alla födda 83 och därefter. En dylik omkodning ser ut enligt följande (jag har dock vänt på ordningen så att de äldsta har den högsta koden, vilket egentligen inte spelar någon större roll men känns något mer logisk, hög ålder/högt värde):

RECODE

**fodar**

(0 thru 75=7) (76 thru 78=6) (79=5) (80=4) (81=3) (82=2) (83 thru highest=1) INTO **fodar\_2** .

EXECUTE .

Vi kan notera att detta skript är uppbyggt på samma sätt som när vi kodar om samma variabel, med ett undantag – denna gång lägger vi till INTO **fodar\_2** efter att vi har specificerat omkodningarna. Skriptet talar om att vi ska koda om en variabel – **fodar** – på ett visst sätt till en ny variabel – **fodar\_2**. Skulle vi ta bort INTO **fodar\_2** kodar vi istället om variabeln **fodar.** Resultatet av omkodningen visas nedan.

1. Födelseår, sju kategorier, elever i gymnasieskolan 1997—1998.[[10]](#footnote-10)

****

Vi ser att grupperna fortfarande är rätt olika i storlek. Två grupper, grupp 1 och grupp 7, utgör mindre än en procent av den totala populationen. Det kan här vara lämpligt att ytterligare aggregera variabeln. Denna gång kan vi utgå från den nya variabeln för att aggregera värdena. Förslagsvis är det lämpligt att slå samman grupp 1 och 7 med närmaste åldersgrupp, 2 respektive 6, vilket ger enligt följande skript:

RECODE

**fodar\_2**

(1 thru 2=1) (3=2) (4=3) (5=4) (6 thru 7=5) INTO **fodar\_3** .

EXECUTE .

Vi erhåller nu följande resultat:

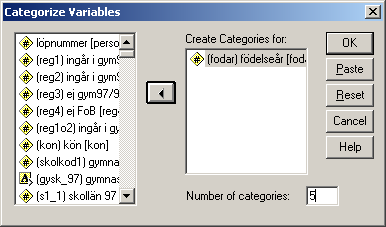
1. Födelseår, fem kategorier, elever i gymnasieskolan 1997—1998.

****

Möjligen skulle man även kunna aggregera grupp 4 och 5 eftersom grupp 5 fortfarande är relativt liten jämfört de andra.

### Användning av Categorize Variable för att automatiskt koda om en variabel i lika stora grupper

Vill vi skapa nya kategorier (variabelvärden) som är så pass lika till antalet individer per kategori kan vi använda oss av funktionen *Categorize Variable*. Gå in på *Transform/Categorize Variable....*



Vi väljer här samma variabel som i exemplet ovan, fodar, och specificerar antalet kategorier till 5. Vi får följande skript:

RANK

VARIABLES = fodar

/NTILES(5)

/PRINT = NO

/TIES = MEAN .

Vi skapar därmed en ny variabel som omfattar fem kategorier. Om vi gör en korstabell ser vi nu att vi faktiskt inte fått fem kategorier utan endast fyra. Pga. den kraftiga dominansen av vissa födelseår slår programmet samman kategori 1 och 2 och vi erhåller endast kategorierna 1, 3, 4 och 5. I fallet med ålder kommer dock inte funktionen *Categorize Variable* till sin fulla rätt. Det är relativt enkelt att, som ovan, själv dela upp variabeln. Om vi däremot ska dela upp en variabel som inkomst med tusentals olika värden kan däremot funktionen *Categorize Variable* vara mycket användbar.

(fodar) födelseår \* NTILES of FODAR Crosstabulation

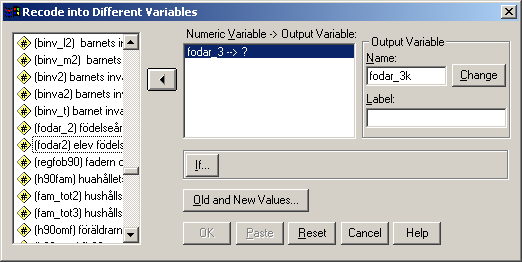
Count

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | NTILES of FODAR |  |  |  | Total |  |
|  |  | 1 | 3 | 4 | 5 |  |  |
| (fodar) födelseår | 0 | 3 |  |  |  | 3 |  |
|  | 10 | 1 |  |  |  | 1 |  |
|  | 28 | 1 |  |  |  | 1 |  |
|  | 32 | 1 |  |  |  | 1 |  |
|  | 34 | 1 |  |  |  | 1 |  |
|  | 38 | 1 |  |  |  | 1 |  |
|  | 41 | 1 |  |  |  | 1 |  |
|  | 42 | 3 |  |  |  | 3 |  |
|  | 43 | 2 |  |  |  | 2 |  |
|  | 44 | 4 |  |  |  | 4 |  |
|  | 45 | 5 |  |  |  | 5 |  |
|  | 46 | 5 |  |  |  | 5 |  |
|  | 47 | 5 |  |  |  | 5 |  |
|  | 48 | 5 |  |  |  | 5 |  |
|  | 49 | 5 |  |  |  | 5 |  |
|  | 50 | 7 |  |  |  | 7 |  |
|  | 51 | 14 |  |  |  | 14 |  |
|  | 52 | 15 |  |  |  | 15 |  |
|  | 53 | 21 |  |  |  | 21 |  |
|  | 54 | 24 |  |  |  | 24 |  |
|  | 55 | 14 |  |  |  | 14 |  |
|  | 56 | 25 |  |  |  | 25 |  |
|  | 57 | 24 |  |  |  | 24 |  |
|  | 58 | 30 |  |  |  | 30 |  |
|  | 59 | 35 |  |  |  | 35 |  |
|  | 60 | 46 |  |  |  | 46 |  |
|  | 61 | 44 |  |  |  | 44 |  |
|  | 62 | 59 |  |  |  | 59 |  |
|  | 63 | 61 |  |  |  | 61 |  |
|  | 64 | 66 |  |  |  | 66 |  |
|  | 65 | 64 |  |  |  | 64 |  |
|  | 66 | 64 |  |  |  | 64 |  |
|  | 67 | 82 |  |  |  | 82 |  |
|  | 68 | 89 |  |  |  | 89 |  |
|  | 69 | 91 |  |  |  | 91 |  |
|  | 70 | 136 |  |  |  | 136 |  |
|  | 71 | 171 |  |  |  | 171 |  |
|  | 72 | 217 |  |  |  | 217 |  |
|  | 73 | 284 |  |  |  | 284 |  |
|  | 74 | 410 |  |  |  | 410 |  |
|  | 75 | 724 |  |  |  | 724 |  |
|  | 76 | 1607 |  |  |  | 1607 |  |
|  | 77 | 4312 |  |  |  | 4312 |  |
|  | 78 | 19441 |  |  |  | 19441 |  |
|  | 79 | 100211 |  |  |  | 100211 |  |
|  | 80 |  | 102591 |  |  | 102591 |  |
|  | 81 |  |  | 99383 |  | 99383 |  |
|  | 82 |  |  |  | 94169 | 94169 |  |
|  | 83 |  |  |  | 815 | 815 |  |
|  | 84 |  |  |  | 15 | 15 |  |
|  | 85 |  |  |  | 4 | 4 |  |
|  | 96 |  |  |  | 3 | 3 |  |
|  | 97 |  |  |  | 2 | 2 |  |
|  | 99 |  |  |  | 1 | 1 |  |
| Total |  | 128431 | 102591 | 99383 | 95009 | 425414 |  |

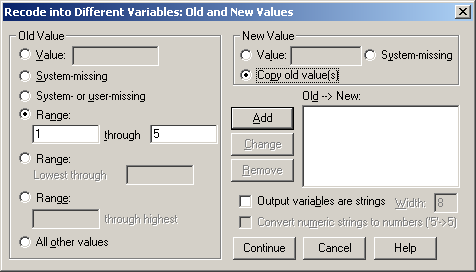
### Skapande av kombinerad variabel

Ibland vill man utnyttja informationen från två variabler i en enda variabel. Vi kan ta kön som exempel. Denna variabel har två värden, man och kvinna. Men detta är ingen uttömmande information om en individ, vi definieras inte uttömmande av vårt kön. Vi har även olika ålder, tillhör olika sociala klasser, har olika nationell härkomst, bor i olika delar av landet, etc. För att skapa en mer komplex kan variabel kan vi utnyttja information från andra variabler. Exempelvis kan vi dela in män och kvinnor efter födelseår.

Detta låter sig göras genom att vi kodar om födelseår till en ny variabel som även innehåller information om kön. För att göra detta behöver vi introducera en ny funktion i omkodningen, villkorssatsen (knappen *if*). Via menyn går vi in på *Transform/Recode/Into Different Variables...* Vi specificerar vilken variabel vi ska koda om, i det här fallet **fodar\_3.** Därefter ställer vi oss i rutan *Output Variabel* och skriver in namnet på den nya variabel, låt säga **fodar\_3k.** Tryck sedan på *Change* och då för vi följande text i mittenrutan: **fodar\_3 🡪 fodar\_3k.**

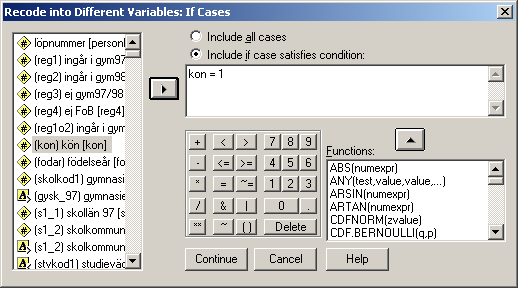


Nästa steg blir att specificera hur värdena ska kodas om. För enkelhets skull väljer vi att koda om värdena 1-5 till samma värden genom att skriva:

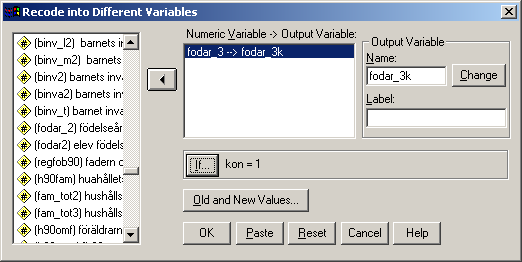


Istället för att specificera värden för *New Value* klickar man på *Copy old value(s)*. Tyck därefter på *Add* och *Continue*.

Nästa steg blir att skapa en villkorssats. Detta görs genom att man klickar på *If*. Man får då upp fönstret *Recode into Different Variables: If Cases*. Här klickar man på *Include if case satisfies condition*. Det blir då möjligt att välja ut variabler i rutan till väster. Markera [kon] kön [kon] och skicka in variabeln till den högra rutan genom att klicka på pilen. Skriv sedan efter kon ”= 1” och tryck på *Continue*.



Vi har nu specificerat att vi endast kodar om de individer som uppfyller kravet att de har värde 1 på variabeln kon. Välj därefter *Ok* om du vill göra omkodningen direkt, eller *Paste* om du vill köra skriptet via syntaxredigeraren.



Om vi väljer Paste får vi följande skript:

DO IF (**kon** = 1 ) .

RECODE

**fodar\_3**

(1 thru 5=Copy) INTO **fodar\_3k** .

END IF .

EXECUTE .

Villkorssatsen uttrycks i SPSS som en *Do if*-sats. Notera att villkoret omsluts av parantes och att det avslutas med punkt. (Ifall man missar punkten kodas variabeln om utan att man tar hänsyn till villkorssatsen.) Villkorssatsen sluts in av en *End if*-sats som talar om att villkorssatsen slutar gälla. Det skriptet säger är att om variabeln kon har värdet 1 så koda om variabel fodar\_3 till fodar\_3k på så sätt att värdena 1 till 5 kopieras.

Variabeln kon har dock även värdet 2 och för att koda om även dessa redigerar vi de skript vi genererat ovan. Vi börjar med att byta ut villkorssatsen till DO IF (**kon** = 2 ) . för att tala om att vi vill endast koda om de individer som har värdet 2. För att det ska vara någon poäng med omkodningen måste vi också ändra värdena som ska omkodas. Värdena 1 till 5 är redan upptagna så vi väljer värdena 6 till 10. Vi får då följande skript:

DO IF (**kon** = 2 ) .

RECODE

**fodar\_3**

(1=6) (2=7) (3=8) (4=9) (5=10) INTO **fodar\_3k** .

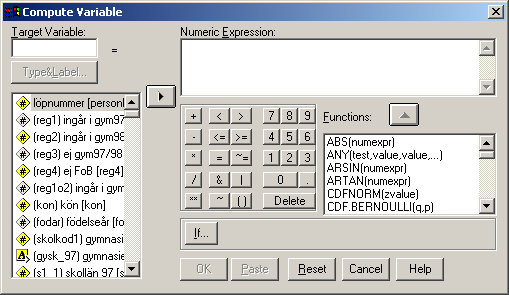
END IF .

EXECUTE .

Samma princip kan också användas för att koda om sociala grupper, så att man skiljer mellan söner och döttrar i läkarfamiljer, civilingenjörsfamiljer, klasslärarfamiljer, etc., eller för att slå samman betyg och social klass (där man exemeplvis kan erhålla grupperna hög social klass/höga betyg, hög social klass/medelhöga betyg, hög social klass/låga betyg, medelklass/höga betyg ...).

### Beräkningar av variabler

Det finns tillfällen då man vill göra beräkningar av variabler och på så sätt skapa nya variabler. För att göra detta går man in på *Transform/Compute*. Man får du upp följande fönster:



Låt säga att vi är intresserade av att få fram det högsta betyg en elev har i något språk. Börja med att skriva in namnet på den nya variabel (den variabel som blir resultatet av beräkningen), exempelvis b98gy\_sp, i rutan *Target Variable*. För att genomföra beräkningen kan man dels själv specificera en formel genom att skriva direkt i rutan *Numeric Expression*, dels använda någon av de specificerade formlerna i rutan *Functions*.

För att få reda på det högsta betyg eleven har i något språk är det enklast att välja funktionen *MAX(value,value,...)*. Denna funktion väljer ut det högsta värde som återfinns bland ett antal variabler. Det är således viktigt att hålla reda på hur variablerna är kodade och att de är kodade på samma vis. I det här fallet är alla betygskoder kodade på följande vis:

1 Ej läst

2 Ej betyg

3 Icke godkänt

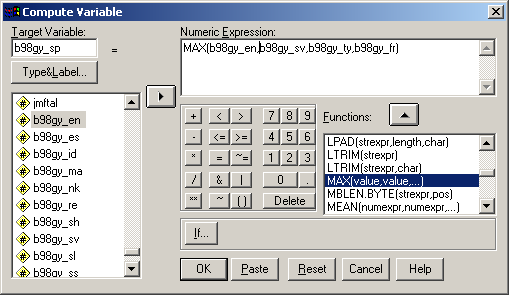
4 Godkänt

5 Väl godkänt

6 Mycket väl godkänt

Det högsta värdet är också det högsta betyget. Skulle vi ha haft en omvänd ordning, där det högsta betyget haft det lägsta värdet kunde vi istället ha valt funktionen *MIN(value,value,...)* som väljer ut det lägsta värdet och erhållit samma resultat.

Efter att ha klickat in funktionen *MAX(value,value,...)* med hjälp av piltangenten i rutan *Numeric Expression* (vi får du uttrycket MAX(?,?) i rutan) väljer vi ut de variabler som vi vill utgå från, i det här fallet b98gy\_en, b98gy\_sv, b98gy\_ty, b98gy\_fr, dvs. betygen i engelska, svenska, tyska och franska. Principen är att vi klickar på den första variabeln b98gy\_en och klickar in den med piltangenten. Det första frågetecknet ersätts nu med variabelns namn. Därefter markerar vi det andra frågetecknet och sedan klickar vi in nästa variabel. Frågetecknet ersätts då med b98gy\_sv. Vi skriver sedan ett kommatecken och klickar in nästa variabel, b98gy\_ty, och upprepar proceduren med variabeln b98gy\_fr. Vi är nu klara med vår formel och har erhållit följande fönster:



Vi kan nu klicka på *OK* eller *Paste* för att köra vår uträkning och därmed skapa en ny variabel. Det skript vi erhåller ser ut på följande sätt:

COMPUTE b98gy\_sp = MAX(b98gy\_en,b98gy\_sv,b98gy\_ty,b98gy\_fr) .

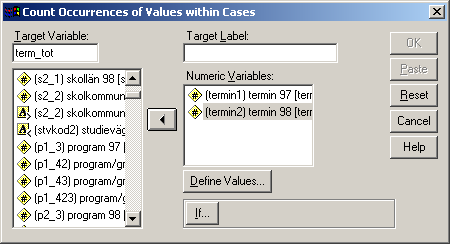
EXECUTE.

Den nya variabeln b98gy\_sp beräknas enligt formeln MAX(b98gy\_en,b98gy\_sv,b98gy\_ty,b98gy\_fr) .

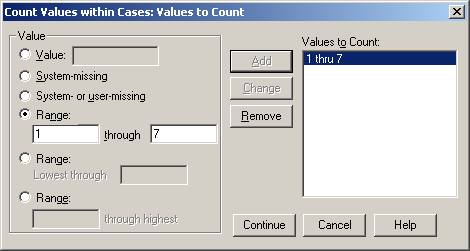
Några andra användbara formler är *Mean* (beräknar medelvärdet), *Sum* (beräknar summan), *Concat* (slår samman strängvariabler, se avsnitt IX.2.10 Skapande av kombinerade strängvariabler), *Lag* (skapar en förskjutning av en serie värden, se kapitel XV.3 Att skapa dataset där individerna utgör endast en rad).

### Räkna antal specifika värden i ett antal variabler

En annan funktion som kan vara användbar är *Count*, som räknar antalet specificerade värden i ett antal variabler. Låt säga att man har ett antal variabler för terminer man läst på gymnasiet. Man vill nu veta hur många träffar som finns för varje individ. Vi kan då använda oss funktionen *Count*, gå in på *Transform/Count...* och erhåller följande fönster:



Vi specificerar att vi vill räkna två variabler, termin1 och termin2. Den nya variabeln döper vi till term\_tot. Därefter specificerar vi vilka värden vi vill räkna. Har vi gjort en frekvenstabell innan vet vi att de värden som finns för termin1 och termin2 är 1 till och med 7. Vi är intresserade av alla dessa värden och använder därför *Range*.



Vi får nu följande skript:

COUNT

**term\_tot** = **termin1 termin2** (1 thru 7) .

EXECUTE .

Ifall vi endast är intresserade av att räkna dem som går på termin 1, dvs. få fram vilka som läser på termin 1 två år i rad kan vi formulera skriptet på följande sätt:

COUNT

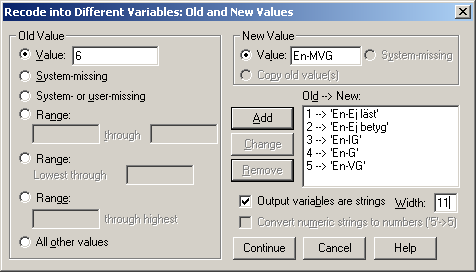
**term\_tot** = **termin1 termin2** (1) .

EXECUTE .

Observera att det endast går att räkna på numeriska variabler. Om man vill räkna antalet värden i strängvariabler måste man först koda om strängvariablerna till numeriska variabler.

### Omkodning av en numerisk variabel till en strängvariabel och vice versa

Av olika anledningar kan det vara lämpligt att koda om en strängvariabel en till numerisk variabel. I första hand rekommenderas att detta görs via Automatisk omkodning (*Automatic Recode*), se nästa avsnitt, men det kan ändå vara värt att känna till hur man gör på ett sätt som tillåter redigering av de värden man ska omkoda. Det går dock *inte* att automatiskt omkoda en numerisk variabel till den strängvariabel. Hur detta görs redovisas här. Vi kan använda oss av exemplet med gymnasiebetyg i ett visst ämne, där betygen specificerats ovan. Via menyn går vi till väga som det beskrivits ovan (*Transform/Recode/Into Different Variables...*)med några få viktiga undantag. När vi fått upp fönstret *Recode into Different Variables: Old and New Values* måste vi specificera att den nya variabeln är en strängvariabel. Detta görs genom att vi markerar *Output variables are strings*. Därefter specificerar vi *width*, dvs. hur många tecken variabeln skall ha. I det här fallet bör vi sätta värdet till minst 11 tecken eftersom värdet 2 omfattar 11 tecken.



Det skript som genereras ser ut enligt följande:

STRING **b98gyent** (A11) .

RECODE

**b98gy\_en**

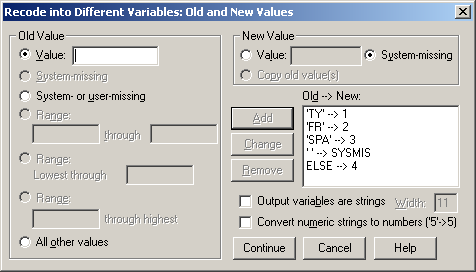
(1='En-Ej läst') (2='En-Ej betyg') (3='En-IG') (4='En-G') (5='En-VG')

(6='En-MVG') INTO **b98gyent** .

EXECUTE .

Observera att först så definieras den nya strängvariabeln, endast nya numeriska variabler skapas automatiskt. Ett värde i en strängvariabel omsluts till skillnad från värden i numeriska variabler av två’. I övrigt överensstämmer skriptet med det som gäller för omkodning av en numerisk variabel till en ny numerisk variabel.

Det går även att koda en strängvariabel till en numerisk variabel. Man går in på samma sätt (*Transform/Recode/Into Different Variables...*) och väljer en variabel som ska kodas om. Vi kan här ta elevers b-språk, bsp\_spr. Vi går sedan in på *Old and New Values* och specificerar de värden vi vill koda om. De flesta elever läser tyska, franska eller spanska. Övriga språk kan slås samman till en kategori. Samtidigt vill vi särskilja dem det inte finns något språk för. Vi specificerar då först det gamla värdet (*Old value*) som ska kodas om genom att skriva TY utan enkla citattecken (SPSS skriver automatisk dessa) och sedan det nya värdet (*New Value*) här 1. Tryck *Add* och upprepa för FR och SP (nytt värde: 2 respektive 3). För att koda de som ej har något värde för språk används mellanslag i *Old value* och System-missing i *New value*. Därefter sätter vi *All other values* till 4.



Följande skript erhålls:

RECODE

**bsp\_spr**

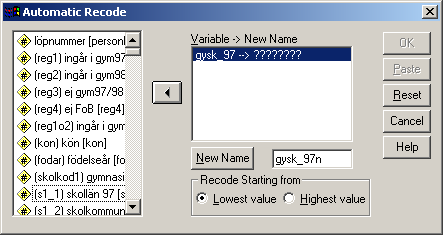
('FR'=2) ('TY'=1) ('SP'=3) (' '=SYSMIS) (ELSE=4) INTO **bsp\_sprt** .

EXECUTE .

Slutligen: för att koda en strängvariabel till en ny strängvariabel måste såväl de gamla värdena som de ny värdena omslutas av enklas citattecken, exempelvis ('FR'=’Franska’) ('TY'=’Tyska’) .

### Automatisk omkodning av strängvariabel till numerisk variabel

En mycket användbar funktion är den automatiska omkodningen från en strängvariabel till en numerisk variabel. Detta är synnerligen praktiskt när man snabbt vill sammanställa ett dataset som går att konvertera till SPAD. För att automatiskt omkoda en variabel gå in på *Transform/Automatic Recode...* . Följande fönster visas:



Börja med att välja den variabel du vill koda om. Klicka in den i *Variable 🡪New Name*-rutan. Skriv namnet på den nya variabeln och klicka på *New Name*. Tryck *OK* eller *Paste*. SPSS kodar nu om strängvariabeln till en numerisk variabel där variablerna läggs i bokstavsordning, dvs. det första värdet enligt bokstavsordning får värdet 1, nästa värde 2 osv. Om det saknas data, dvs. förekommer blanka celler ges dessa värdet 1, och därefter ges det första värdet enligt bokstavsordning värdet 2, osv. Man kan även sortera åt andra hållet genom att klicka på *Recode Starting from* : *Highest value*. Därefter skapar SPSS automatiskt etiketter på så sätt att värdet 1 får en identisk etikett med det första värdet i alfabetisk ordning. Om vi kodar om skolornas namn, variabel gysk\_97, till en ny variabel, gysk\_97n, erhåller vi följande ”översättning”:

GYSK\_97 GYSK\_97N (gysk\_97) gymnasieskola 97, text

Old Value New Value Value Label

1

Abb Industrigymnasium 2 Abb Industrigymnasium

Abf, Norrköpings Konstskola 3 Abf, Norrköpings Konstskola

Agnebergsgymnasiet 4 Agnebergsgymnasiet

Ale Gymnasium 5 Ale Gymnasium

Aleholmsskolan 6 Aleholmsskolan

Alléskolan 7 Alléskolan

Allhamraskolan 8 Allhamraskolan

Almers Gymnasium 9 Almers Gymnasium

Alnängsskolan 10 Alnängsskolan

Det skript som skapas ser ut enligt följande:

AUTORECODE

VARIABLES=**gysk\_97** /INTO **gysk\_97n**

/PRINT.

### Automatisk koncentration värdena i en numerisk variabel

Att göra det omvända, att automatiskt koda om en numerisk variabel till en strängvariabel, är tyvärr inte möjligt. Den som försöker detta uppnår dock en annan värdefull effekt, som är användbar för att skapa ett dataset som lätt går att konvertera till SPAD. Numeriska variabler som innehåller tomma värden eller ”luckor” koncentreras av SPSS om vi väljer automatisk omkodning av en numerisk variabel. Vi gör som ovan men istället för att välja en strängvariabel väljer vi en numerisk variabel och specificerar en ny variabel. Låt oss denna gång ta skollän (s1\_1) som exempel. Den översättning vi nu får ser ut:

S1\_1 S1\_1T (s1\_1) skollän 97

Old Value New Value Value Label

1 1 Gy-Stockholms län

3 2 Gy-Uppsala län

4 3 Gy-Södermanlands län

5 4 Gy-Östergötlands län

6 5 Gy-Jönköpings län

7 6 Gy-Kronobergs län

8 7 Gy-Kalmar län

9 8 Gy-Gotlands län

10 9 Gy-Blekinge län

12 10 Gy-Malmöhus län

13 11 Gy-Hallands län

14 12 Gy-Göteborgs och Bohus län

15 13 Gy-Älvsborgs län

16 14 Gy-Skaraborgs län

17 15 Gy-Värmlands län

18 16 Gy-Örebro län

19 17 Gy-Västmanlands län

20 18 Gy-Kopparbergs län

21 19 Gy-Gävleborgs län

22 20 Gy-Västernorrlands län

23 21 Gy-Jämtlands län

24 22 Gy-Västerbottens län

25 23 Gy-Norrbottens län

Vi ser att medan den gamla variabeln (s1\_1) har värden från 1 till 25 varav två ej finns (2 och 11) skapas en ny variabel (s1\_1t) med 23 värden och inga tomma värden.

Det skript som skapas är identiskt med det som vi skapade ovan för strängvariabler:

AUTORECODE

VARIABLES=**s1\_1** /INTO **s1\_1t**

/PRINT.

### Skapande av kombinerade strängvariabler

Ett effektivt sätt att skapa mer komplexa variabler som kombinerar informationen i flera variabler är att slå samman flera strängvariabler till en ny strängvariabel. Detta åstadkoms antingen genom att gå in på *Transform/Compute* och välja funktionen *Concat* eller genom skrivande av skript (se ovan avsnitt III.2.1.1 Infogande av ny variabel för hur man gör när man skapar en ny variabel via menyn, se nedan för en beskrivning av hur skriptet skrivs). För att resultatet skall bli lyckat bör man dock tänka på några saker.

* Det underlättar att använda korta textsträngar i varje variabel eftersom den sammanslagna variabeln då inte får så långa värden.
* Det är praktiskt att alla värden i en variabel har samma antal tecken
* Det underlättar läsningen av de nya värdena om man separerar de olika variablerna med ett bindestreck eller dylikt.

Ett exempel när det är lämpligt att använda denna funktion är när man ska slå samman gymnasieskola och utbildning. Börja med att skapa en strängvariabel för skolan. Använd förslagsvis tre tecken för att namnge skolan samt ett bindestreck, dvs. totalt fyra tecken. Hur många tecken som används till utbildningen är mindre väsentligt så länge vi inte ska lägga på ytterligare information. Vad gäller utbildningarna kan man använda sig av etablerade förkortningar av program och grenar såsom EP för Estetiska programmet och NVNA för det Naturvetenskapliga programmets naturvetenskapliga gren.

Låt säga att vi skapat en strängvariabel sk98t för skolorna som omfattar tre tecken för skolan samt ett tecken för bindestreck, och en strängvariabel för utbildningen, utb98t, som omfattar 6 tecken. För att slå ihop dessa variabler behöver vi först skapa en ny variabel som omfattar minst 10 tecken (4+6 tecken, dvs. det antal tecken som ingår i de variabler vi ämnar slå samman). Detta gör vi genom att skriva

STRING **skut98** (A10) .

Därefter skriver vi

COMPUTE **skut98** = CONCAT(**sk98t**,**utb98t**) .

EXECUTE.

Vi har nu fått en ny variabel som kombinerar skola och utbildning. Denna kan vi sedan göra om till en numerisk variabel (se ovan avsnitt IX.2.8 Automatisk omkodning av strängvariabel till numerisk variabel för grundprinciperna och nedan Avsnitt XII.3 Att omkoda strängvariabler till numeriska variabler och använda värden i strängvariabeln till etiketter i den numeriska ” för hur man kan göra detta med hjälp av Excel och Word)

### Några tips om omkodning och konstruktion av ett dataset

Generellt gäller att det är bra att ha ordning i sitt dataset. Enklast är att låta variablerna placeras enligt någon specifik logik, exempelvis om man har enkätdata kan man välja den ordning frågorna kommer i enkäten. När jag bygger upp dataset utifrån SCB-data försöker jag att så långt som möjligt hålla samman de olika registren jag hämtat data från. Exempelvis kan ett register baserat på FoB-ar byggas upp enligt följande:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lpnr | FoB-80  originaldata | | | FoB-80  bearb. data | | | FoB-85  originaldata | | | FoB-85  bearb. data | | | FoB-90  originaldata | | | FoB-90  bearb. data | | | FoB-tot  bearb. data | | |
|  | Ba | Mo | Fa | Ba | Mo | Fa | Ba | Mo | Fa | Ba | Mo | Fa | Ba | Mo | Fa | Ba | Mo | Fa | Ba | Mo | Fa |
| Ind1 |  | | |  | | |  | | |  | | |  | | |  | | |  | | |
| Ind2 |  | | |  | | |  | | |  | | |  | | |  | | |  | | |
| Ind3 |  | | |  | | |  | | |  | | |  | | |  | | |  | | |
| Ind4 |  | | |  | | |  | | |  | | |  | | |  | | |  | | |

Här ges naturligtvis en rad andra möjligheter. Jag kan lägga alla originaldata första och sedan lägga bearbetade variabler, jag kan dela upp det så att barnets uppgifter (original och bearbetade) kommer först (FoB-80, FoB-85 och FoB-90), därefter mammans och sedan pappans.

Notera här att jag skiljer noga mellan originaldata och bearbetade data. Till originaldata ränkas exemepelvis de data som SCB levererar eller egna enkätdata. Var väldigt noga med att alltid behålla originaldata, de utgör ju själva grunden för alla kommande bearbetning. Blir det fel någonstans är det viktigt att kunna gå tillbaka till originaldata för att kontrollera.

Ett tips är att tydligt markera vilka variabler som är besläktade med varandra. Om jag har en originalvariabel som skolkod för gymnasieskolor 1997, döpt till sk\_kod97, kan jag ur denna variabel härleda en rad olika variabler. Jag kan skapa en strängvarabel med skolans namn i fulltext, gysk97t, en strängvariabel med endast fyra bokstäver för skolan, gysk97t2, en variabel som skiljer ut skollänet, gyla97, en för skolkommunen, gyko97, etc.

Genom att skriva skript kan man lätt koda om ett stort antal variabler på samma gång. För det första kan man koda om ett stort antal variabler i samma kommando förutsatt att samma värden ska kodas om. Låt säga att vi har en massa dikotoma variabler med värdet 1 och 2 samt Missing. Vi vill nu göra värdet Missing till 3. Då kan vi använda oss av följande omkodningsprincip.

RECODE

**var1 var2 var3 var4 var5 var6**

(1 thru 2=COPY) (SYSMIS=3) INTO **var1a var2a var3a var4a var5a var6a** .

EXECUTE .

Observera att det är viktigt att ordningen är rätt. Variabeln var1 kodas om till var1a, var2 till var2a, etc. Det måste också vara samma antal variabler före och efter specifikationen av hur värdena ska kodas om.

Det går rätt snabbt att komma upp i ett stort antal variabler. Det är med andra ord viktigt att ha ordning i sin fil. Placera därför variablerna i arket på ett logiskt sätt, så att alla variabler som berör ett visst område är placerade bredvid varandra och har likartade namn. Tycker man det är svårt att leta reda på de variabler man vill analysera kan man *Utilities/Define Set....* och skapa ett delset av variabler som man vill se i de fönster man arbetar med. Gå därefter in på *Utilities/Use Sets....* och välj det set av variabler du skapat. De övriga variablerna försvinner då inte från datasetet men de visas inte i de menyer som kommer upp. Du kan lätt ta tillbaka de gömda variablerna genom att ändra inställningarna i *Utilities/Use Sets....*

### Några tips för att koda om SPSS-data för att passa korrespondensanalyser i SPAD

För att förbereda en SPSS-fil för att användas i SPAD krävs att man tänker på ett antal saker:

1. **Skapa variabler med 4 till 10 variabelvärden (kategorier)**. Korrespondensanalysen fungerar bäst om man har ungefär samma antal variabelvärden per variabel. Idealiskt är antal variabelvärden mellan 4 och 10. Kön, som endast har två variabelvärden, utgör här ett problem, men detta kan avhjälpas genom att man slår samman denna variabel med en annan, se nedan.
2. **Skapa kategorier (variabelvärden) som är ungefär lika stora vad gäller antal individer**. Korrespondensanalysen syftar till att lyfta fram skillnader i materialet. Metoden är därför mycket känslig för små grupper, som utgör endast någon procent av det totala antalet individer. Råkar dessa individer dela en annan egenskap intar de lätt mycket extrema positioner i en graf. För att undvika detta bör man skapa variabelvärden eller variabelkategorier som omfattar ungefär lika stora antal individer. En variabel med 5 variabelvärden bör ha ca 20 % av individerna i varje variabelkategori (5/100) och en variabel med 10 variabelvärden ca 10 % (10/100).
3. **Gör fler olika varianter av samma ursprungsvariabel, ex. med 4, 7, 10 variabelvärden**. Det är svårt att på förhand räkna ut vilken indelning av en variabel som är mest effektiv i att särskilja individerna. Ett tips är därför att utifrån en sprcifik originalvariabel skapa ett antal olika variabler med olika antal variabelvärden och som dessutom kan delas in efter olika principer. Variabeln skollän kan exemeplvis kategoriseras efter en geografisk princip, dvs. att län som ligger nära varandra slås samman, och efter en urbaniserings- och näringsprincip, dvs. att storstadslän skiljs från jordbrukslän och från glesbygdslän.
4. **Slå samman variabler till sammansatta variabler**. Ovan nämndes att variabler med få variabelvärden ger problem för korrespondensanalysen. Detta kan avhjälpas genom att man skapar sammansatta variabler som kombinerar informationen från flera olika variabler. Man kan exempelvis slå samman kön med och medelbetyg eller social klass. Detta ger också en helt annan socologisk tolkning. Vi talar då inte om endast pojkar och flickor eller elever med höga och låga betyg, utan om pojkar med höga betyg, flickor med höga betyg respektive pojkar med låga betyg och flickor med låga betyg.

# Filter

När vi utgår från totalpopulationer, exempelvis samtliga elever i gymnasieskolan, använder vi ofta väldigt stora populationer. Totalpopulationen kan dock delas in i en rad olika delpopulationer såsom alla elever i årskurs 1, 2 eller 3, alla elever i ett visst län eller en viss kommun, alla elever på vissa typer av program eller skolor. Ibland är det intressant att bara analysera någon specifik delpopulation. Detta kan göras genom att vi filtrerar vårt dataset. För att enkelt göra detta kan det vara bra att först ha skapat en filtervariabel (detta kan förvisso också göras av programmet, men jag föredrar att göra det själv).

Låt säga att vi är intresserade av endast elever som går i Stockholms län 1997. Vi skapar då en filtervariabel som skiljer ut dessa elever från övriga. Ett dylikt skript ser ut enligt följande:

RECODE

**s1\_1**

(1=1) INTO **f\_st97** .

EXECUTE .

Skriptet talar om att vi kodar om variabeln s1\_1 (skollän 97) till en ny variabel f\_st97 (jag brukar namnge alla filtervariabler med prefixet ”f\_” för att markera att det är en filtervariabel) på så sätt att alla som har värdet 1 (dvs. gymnasieskola i Stockholms län) får värdet 1 i den nya variabeln. Övriga värden, dvs. övriga skollän, ges inget värdet i den nya variabeln.

Vi kan även filtrera ut de elever som endast går i årskurs 2 i Stockholms län 1997. Det skriptet blir:

DO IF (**termin1**=3) .

RECODE

**s1\_1**

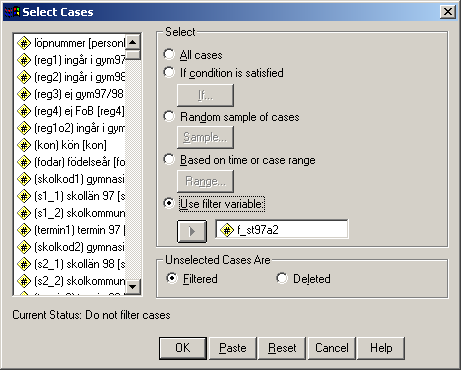
(1=1) INTO **f\_st97a2** .

END IF .

EXECUTE .

Med andra ord: om eleven går tredje terminen 1997 (dvs. andra årskursen) så koda om alla som går i en gymnasieksola i Stockholms län till en ny variabel. Vill vi koda om alla som går i årskurs 2 och 3 ersätter vi *Do-if*-satsen med DO IF (RANGE(**termin1**,3,5)) .

För att filtrera ut delpopulationen går vi in *Data/Select cases...* . Välj sedan *Use filter variable* och markera den variabel som vi vill använda som filtervariabel, här f\_st97a2. Tryck sedan *OK* eller *Paste* för att filtrera datasetet.



Skriptet för att filtrera enligt en viss variabel ser ut enligt följande:

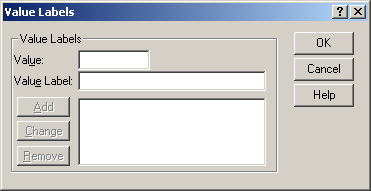
USE ALL.

FILTER BY **f\_st97a2** .

EXECUTE .

# Etiketter på värden

För att veta vad de olika värdena på en variabel egentligen står för är det lämpligt att sätta dit etiketter. Detta kan åstadkommas via redigering av variabelarket. Klicka på Values för den variabeln du vill sätta in etiketter på värdena. Klicka sedan på den grå knappen till höger i rutan. Du får då ett fönster:



Skriv en 1:a i *Value* och den text som du vill ska förknippas med värdet i *Value Label*. Klicka sedan på *Add*. Skriv sedan 2 i Value och en ny text i *Value Label*. Klicka sedan på *Add*. Upprepa detta till du har satt in alla etiketter. Tryck på *OK*. Om du nu gör exempelvis en frekvenstabell kommer värdena att ersättas med dina etiketter.

Ibland har man dock många etiketter för en variabel, eller väldigt många variabler, och då kan det vara bättre att använda sig av syntaxredigeraren för att specificera etiketterna. Detta görs relativt enkelt genom följande syntax:

VALUE LABELS

**kon**

1 'Man' 2 'Kvinna' .

Observera att det inte behövs något EXECUTE . efter att man har specificerat etiketterna. Dock krävs en punkt.

Ifall man har flera variabler med samma etiketter går det bra att köra dessa i samma skript enligt modellen:

VALUE LABELS

**var1 var2 var3 var4 var5**

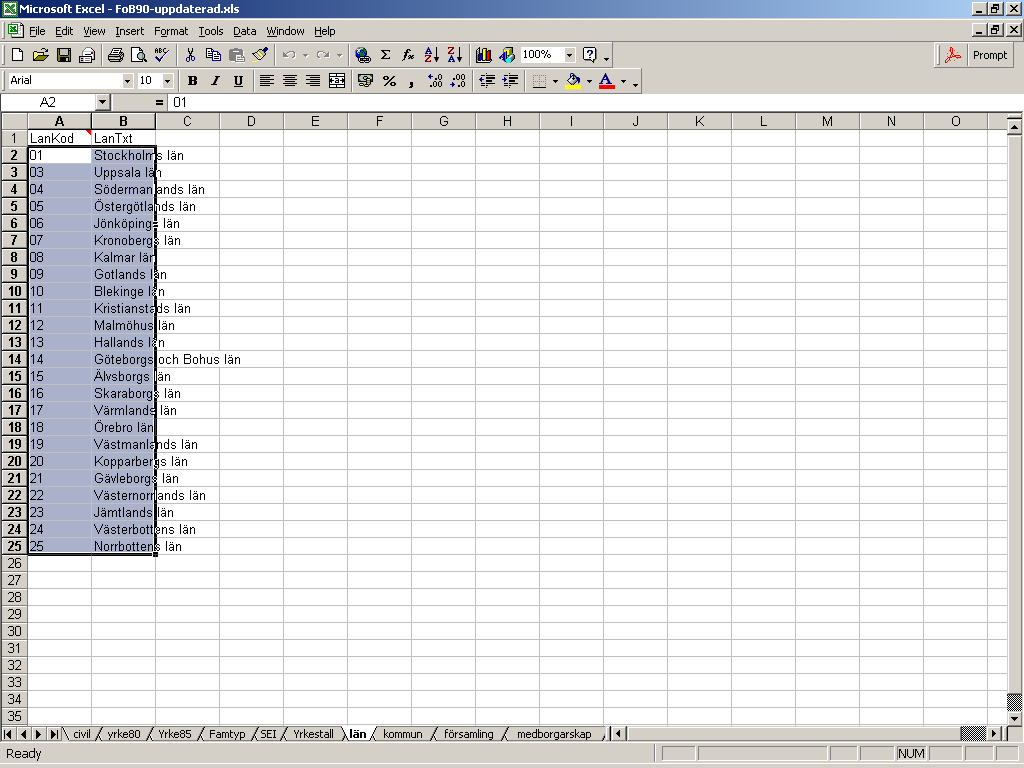
1 '**Etikett1**' 2 '**Etikett2**' 3 ’**Etikett3**’ .

# Användande av Word och Excel för skapande av skript

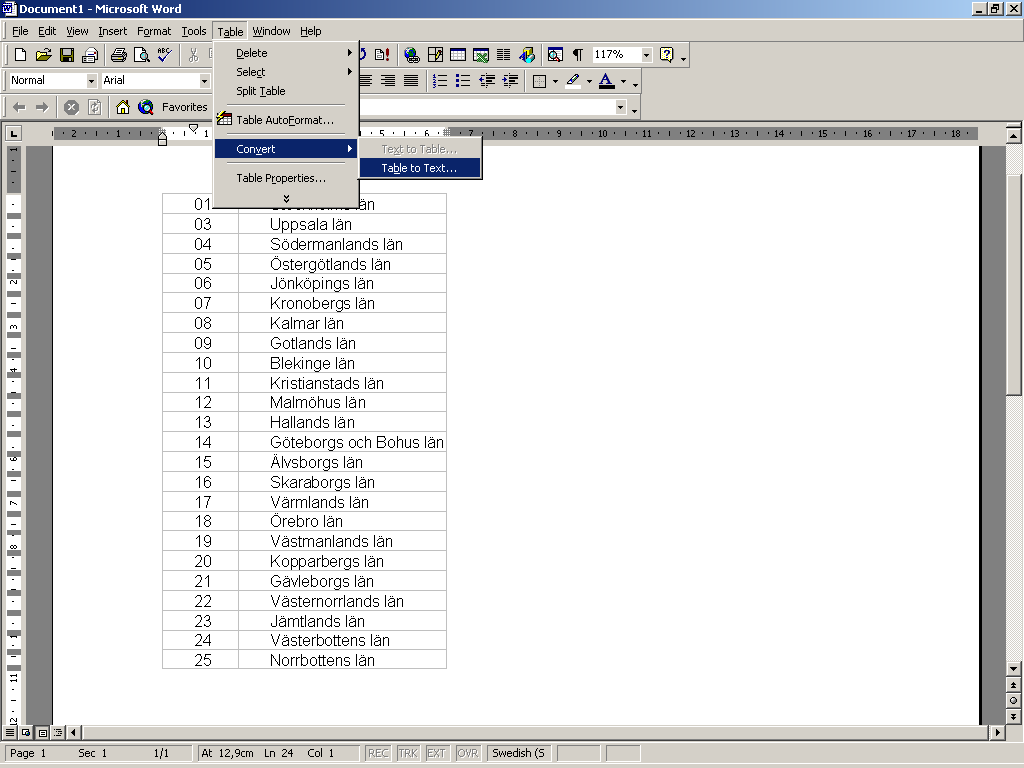
När man jobbar med stora datafiler och har många variabler med hundratals olika värden (exempelvis samtliga grundskolor i Sverige eller samtliga församlingar eller kommuner) kan det vara tidsbesparande och ofta helt nödvändigt att använda sig av hjälpmedel för att underlätta omkodning och etiketterande av variabler. Ett sätt är att använda sig av Excel och Word för att skapa skript. Även om detta är mest tidsbesparande vad gäller variabler men många värden kan principerna även användas för att koda om variabler med få värden – vi kommer här att utgå från det län en gymnasieskola är belägen i som ett exempel (25 värden). Först ska vi sätta etiketter på värdena, därefter koda om till en strängvariabel.

## Att skapa etiketter

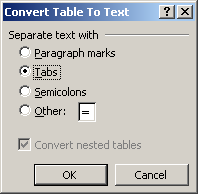
I det här fallet kommer våra data från SCB och de levereras då i en form där de flesta variabler är numeriska variabler. Dessutom får man en metadatafil i Excel-format som talar om vad de olika värdena betyder. Detta gör att man kan använda metadatafilen för att skapa etiketter, dessa finns ju redan i digital form i Excelarket och man slipper således skriva alla etiketter manuellt. Vi börjar med att leta reda på information om länskoderna i Excelarket:



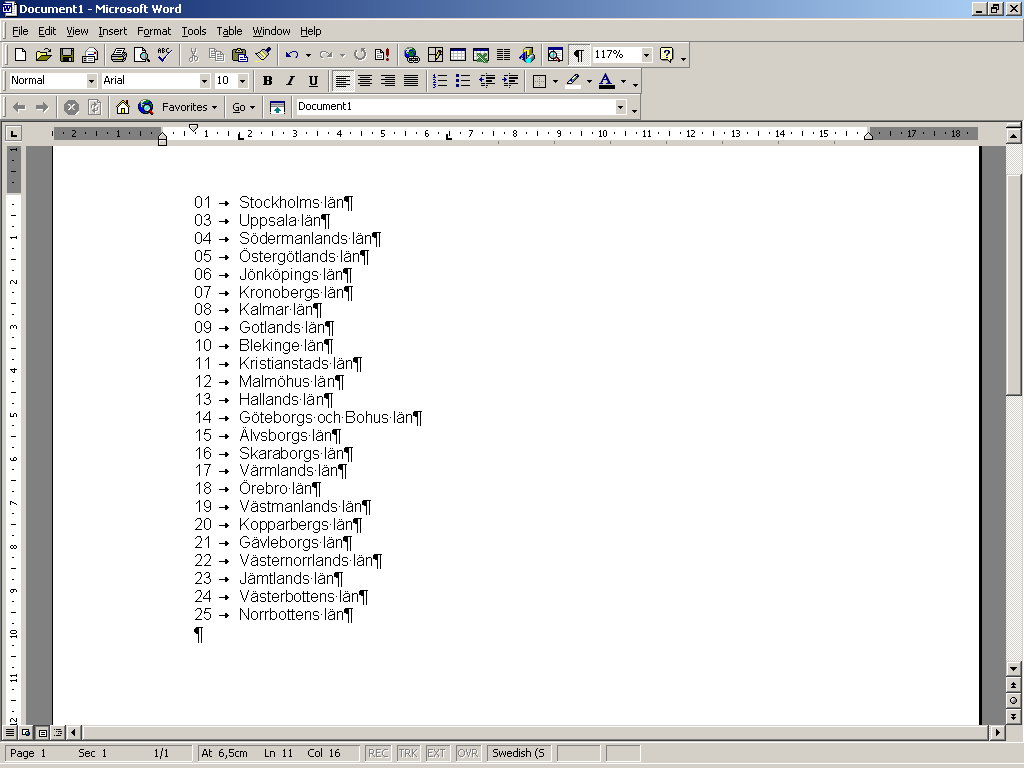
Därefter markerar vi samtliga värden och samtliga förklarande text och kopierar detta utsnitt. Nästa steg är att klistra in allt i Word. Vi får då en tabell. Ställ markören i tabellen. Gå in på *Tabel/Convert/Tabel to Text...*.



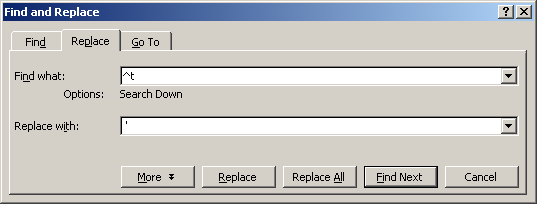
Följande fönster uppkommer:



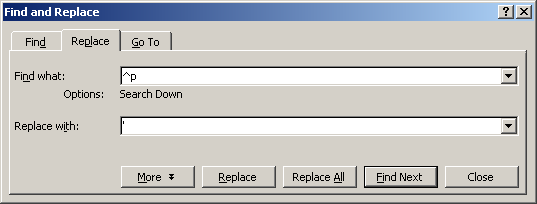
I det här fallet kan vi separera texten med tabbar. Tryck *OK*. Tabellen omvandlas då till ren text. De två tidigare kolumnerna blir nu separerade av ett tabbtecken. Detta framgår tydligt när vi väljer att visa alla tecken, tryck på radbrytningssymbolen, i mitt Wordprogram, bredvid inställningen av storleken på dokumentet. Vi ser då att varje rad har samma uppbyggnad. Först kommer siffran, därefter en tabb, sedan den förklarande texten (länet) och avslutningsvis en radbrytning, se bild nedan. Detta kan vi utnyttja för att redigera texten.



Markera all text <ctrl+a>. Välj därefter ersätt <ctrl+h>[[11]](#footnote-11) . Följande fönster uppkommer:



Det vi ska ersätta först är tabbarna. Skriv ”^t”, vilket betyder tabb, i det övre fältet (*Find what*). Skriv sedan mellanslag och enkelt citattecken i det undre fältet (*Replace with*). Tryck därefter på *Replace All*. Nästa steg blir att ersätta radbrytningen. I det övre fället skrivs ”^p”, vilket betyder radbrytning, och i det nedre fältet enkelt citattecken och mellanslag. Tryck därefter på *Replace All*.



Vi får då en text som ser ut enligt följande:

01 ‘Stockholms län’ 03 ‘Uppsala län’ 04 ‘Södermanlands län’ 05 ‘Östergötlands län’ 06 ‘Jönköpings län’ 07 ‘Kronobergs län’ 08 ‘Kalmar län’ 09 ‘Gotlands län’ 10 ‘Blekinge län’ 11 ‘Kristianstads län’ 12 ‘Malmöhus län’ 13 ‘Hallands län’ 14 ‘Göteborgs och Bohus län’ 15 ‘Älvsborgs län’ 16 ‘Skaraborgs län’ 17 ‘Värmlands län’ 18 ‘Örebro län’ 19 ‘Västmanlands län’ 20 ‘Kopparbergs län’ 21 ‘Gävleborgs län’ 22 ‘Västernorrlands län’ 23 ‘Jämtlands län’ 24 ‘Västerbottens län’ 25 ‘Norrbottens län’

För att färdigställa skriptet behöver vi nu bara skriva till:

Value Labels

**s1\_1**

framför texten och sätta en punkt i slutet. Observera att det ibland kan komma med överflödiga enkla citattecken i den redigerade texten i Word och att det är viktigt att kolla att texten är rätt i början och slutet.

Det färdiga skriptet blir:

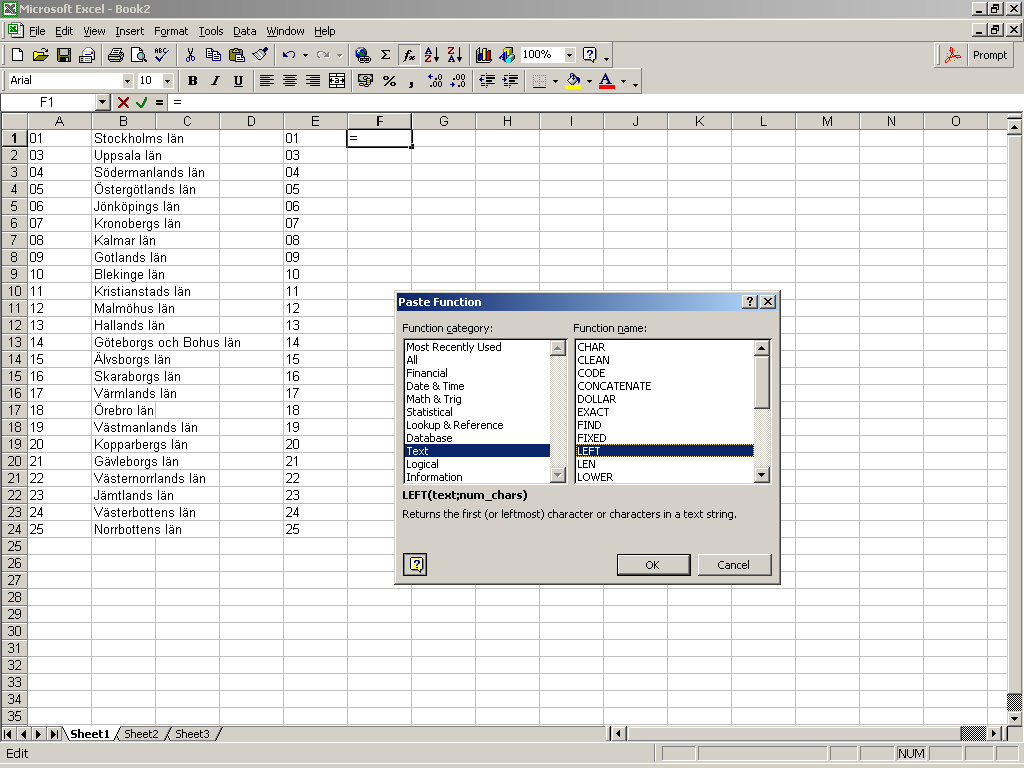
Value Labels

**s1\_1**

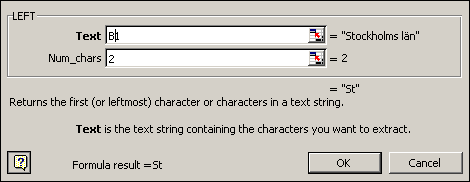
01 ‘Stockholms län’ 03 ‘Uppsala län’ 04 ‘Södermanlands län’ 05 ‘Östergötlands län’ 06 ‘Jönköpings län’ 07 ‘Kronobergs län’ 08 ‘Kalmar län’ 09 ‘Gotlands län’ 10 ‘Blekinge län’ 11 ‘Kristianstads län’ 12 ‘Malmöhus län’ 13 ‘Hallands län’ 14 ‘Göteborgs och Bohus län’ 15 ‘Älvsborgs län’ 16 ‘Skaraborgs län’ 17 ‘Värmlands län’ 18 ‘Örebro län’ 19 ‘Västmanlands län’ 20 ‘Kopparbergs län’ 21 ‘Gävleborgs län’ 22 ‘Västernorrlands län’ 23 ‘Jämtlands län’ 24 ‘Västerbottens län’ 25 ‘Norrbottens län’ .

## Att omkoda från numerisk variabel till strängvariabel

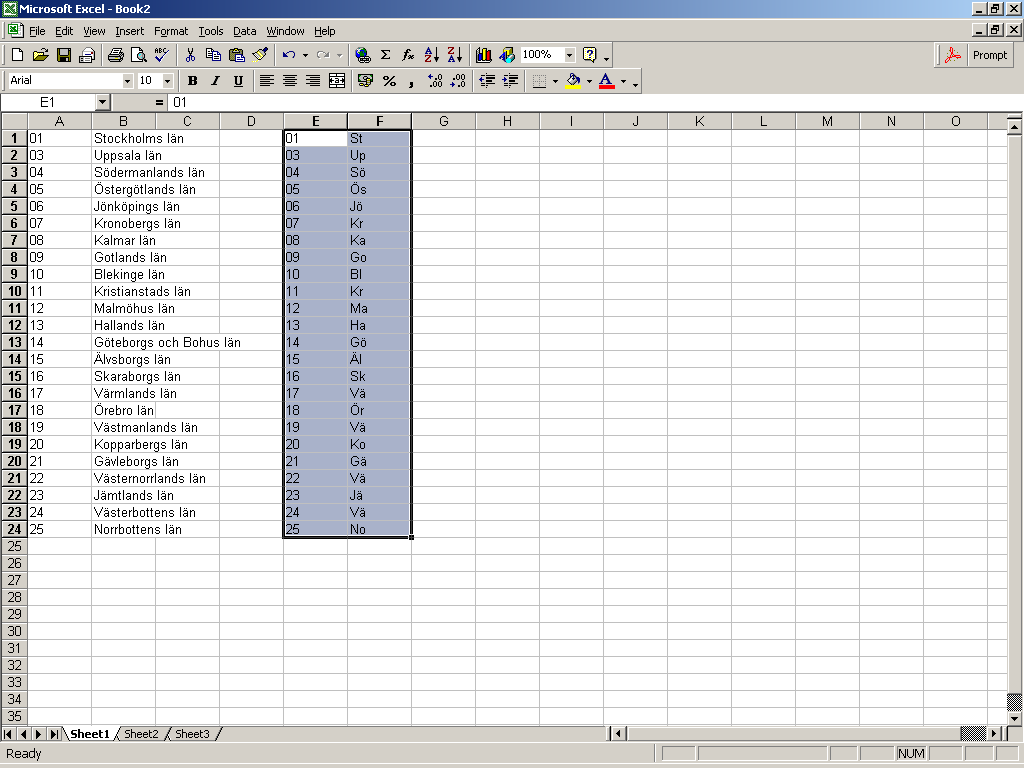
För att koda om en numerisk variabel till en strängvariabel börjar vi med att redigera i Excelarket. I det här fallet kan man tänka sig att skapa en strängvariabel med två tecken som ersätter de numeriska värdena. För enkelhets skull kan vi ta de två första bokstäverna i länets namn. Detta gör vi på följande sätt: Kopiera raden med värden och sätt den till höger om värdena och den förklarande texten (länen) förslagsvis i kolumn E. Observera att det är bäst att sätta klistra in värdena på samma rad, dvs. rad 1 i vårt fall. Ställ markören i cellen F1. Gå därefter in på *Insert/Funcion* och välj funktionskategori *Text* och funktion *Left*, se bild nedan.



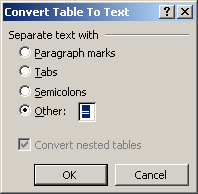
Ett nytt fönster kommer upp.



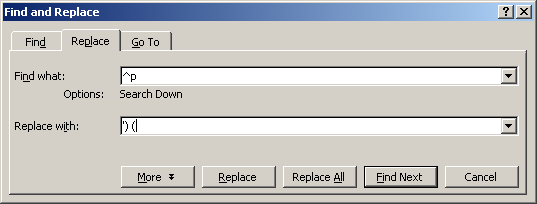
Skriv in B1 i rutan *Text* och sätt värdet av *Num\_chars* till 2 (vill man ha de tre första bokstäverna från vänster skriver man 3 här). Tryck *OK*. Markera därefter cellen F1 och kopiera den. Klistra in den i F2 till F24. (Alternativt markera cellerna F1 till F24 och använd <ctrl+d>.) Följande resultat har nu uppnåtts:



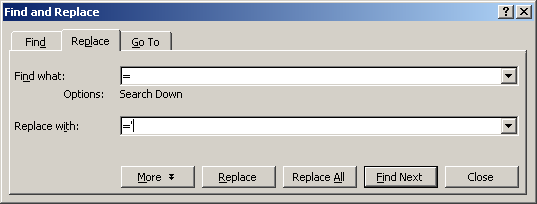
Markera därefter E1 till F24, kopiera och klistra in i Word. Ställ dig i tabellen och gå in på *Tabel/Convert/Tabel to Text...*. Välj denna gång att separera text med *Other* och välj ”=”:



Ersätt sedan text. Välj denna gång ”^p” och ersätt radbrytningen med enkelt citattecken högerparantes mellanslag vänsterparantes ”’) (”. Tryck på *Replace All*.



Ersätt sedan ”=” med ”=’”. Tryck på *Replace All*.



Du har nu fått följande text:

01=‘St’) (03=‘Up’) (04=‘Sö’) (05=‘Ös’) (06=‘Jö’) (07=‘Kr’) (08=‘Ka’) (09=‘Go’) (10=‘Bl’) (11=‘Kr’) (12=‘Ma’) (13=‘Ha’) (14=‘Gö’) (15=‘Äl’) (16=‘Sk’) (17=‘Vä’) (18=‘Ör’) (19=‘Vä’) (20=‘Ko’) (21=‘Gä’) (22=‘Vä’) (23=‘Jä’) (24=‘Vä’) (25=‘No’) (

Flytta först den sista högerparentesen till början av texten (alla värden måste vara omslutna av parenteser). Lägg därefter i början till

RECODE

**s1\_1**

och lägg i slutet till:

into **s1\_1t** .

EXECUTE .

Vi ska nu skapa en ny strängvariabel. Då krävs det också att vi skapar den nya variabeln i skriptet. Skriptet måste således börja med:

STRING **s1\_1t** (A2) .

(A2 står för att det är en strängvariabel med 2 tecken, man kan specificera allt mellan 1 och 255 tecken.) Hela skriptet blir nu:

STRING **s1\_1t** (A2) .

RECODE

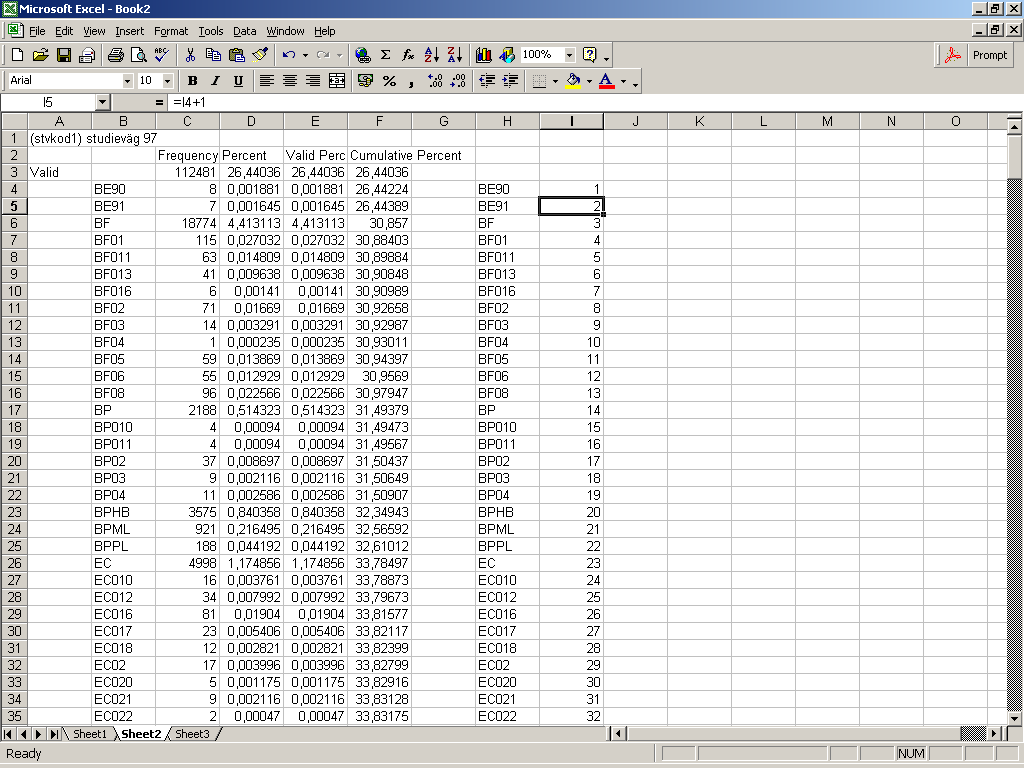
**s1\_1**

(01=‘St’) (03=‘Up’) (04=‘Sö’) (05=‘Ös’) (06=‘Jö’) (07=‘Kr’) (08=‘Ka’) (09=‘Go’) (10=‘Bl’) (11=‘Kr’) (12=‘Ma’) (13=‘Ha’) (14=‘Gö’) (15=‘Äl’) (16=‘Sk’) (17=‘Vä’) (18=‘Ör’) (19=‘Vä’) (20=‘Ko’) (21=‘Gä’) (22=‘Vä’) (23=‘Jä’) (24=‘Vä’) (25=‘No’) into **s1\_1t** .

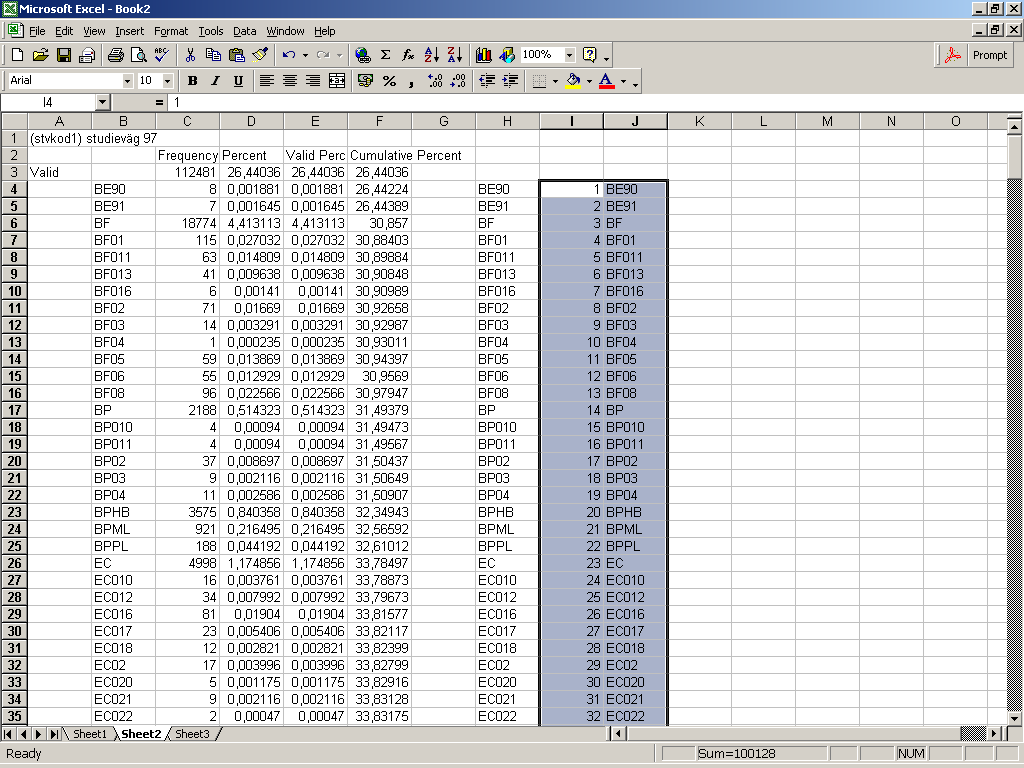
EXECUTE .

## Att omkoda strängvariabler till numeriska variabler och använda värden i strängvariabeln till etiketter i den numeriska variabeln

I avsnitt IX.2.10 Skapande av kombinerade strängvariabler skapade vi kombinerade strängvariabler. När vi skall göra korrespondensanalyser på våra data måste alla strängvariabler göras om till numeriska variabler. Detta görs enklast via Automatisk omkodning se avsnitt IX.2.8 Automatisk omkodning av strängvariabel till numerisk variabel. Om man däremot vill ändra något i omkodningen kan det vara enklare att med hjälp av Excel och Word kombinerar de två tekniker som beskrivs ovan. Börja med att dra ut en frekvens på den kombinerade strängvariabeln (här har vi dock valt en annan strängvariabel, studieväg 1997, stvkod1). Kopiera tabellen och klipp in den i Excel. Kopiera strängvärdena och klistra in dem i en ny kolumn. Därefter skall vi skapa en nummerserie som går från 1 och uppåt så att vi får lika många en serie som motsvarar antalet värden strängvariabeln omfattar. Detta görs på så sätt att vi ställer oss i rutan till höger om det första strängvärdet. Skriv 1 i den rutan. Gå därefter en ruta ner, rakt under 1:an. Skriv i denna ruta ”= [namnet på cellen med värdet 1] + 1”, exempelvis ”=I4 + 1”. Du får nu värdet 2 i cellen. Kopiera cellen med värdet 2 (cell I5 i exemplet nedan) och klistra in denna i resterande celler under cellen så att samtliga strängvärden får ett värde. Du har nu erhållit följande:



För att för skapa en numerisk variabel av detta följer du stegen i Avsnitt III.2.1.1 Infogande av ny variabel ovan. För att sedan skapa etiketter kopierar du värdena i kolumn H (i det här exemplet, se bilden ovan, värden kan finnas i en annan kolumn beroende på var du klistrat in dem från början) och placerar dem i kolumn J, se bild nedan. Om du vill ändra ordningen på hur variablerna kodas eller ändra texten går detta att göra i Excel.



Nu följer du stegen i Avsnitt XI Etiketter på värden ovan för att skapa etiketter.

# Aggregering och transponering av dataset

## Aggregering

Att aggregera data är att på något finurligt sätt slå samman flera observationer till *en* observation. Om vi till exempel våra observationer, det vill säga rader i tabellen, är individer, och vi har data om hemkommun, skola och program, kan vi slå samman alla individer som läser på samma skola. Vi får då en tabell med en rad för varje skola. För att det skall vara någon poäng med en dylik sammanslagning måste vi ha någon intressant information om *skolorna*. Kommandot *Data/Aggretate...* låter oss skapa sådan information, vilket i detta fall skulle kunna vara antalet elever som läser på varje skola.

Kommandot *Aggretate* har tre delar:

1. Filnamnet där den nya aggregerade tabellen skall sparas.
2. En beskrivning av principen för hur observationer (rader) skall slås samman.
3. En beskrivning av vilken typ av information som skall finnas i den nya tabellen.

I beskrivningen av hur kommandot fungerar kommer jag hålla mig till hur man skriver i syntaxfönstret. Man kan även välja aggregering under menyn ”data”. Förhoppningsvis kommer beskrivningen här av hur kommandot är uppbyggt att underlätta även för de som väljer att arbeta med menyerna istället för syntaxfönstret. Jag kommer här inte att ge någon helt generell beskrivning av kommandot utan gå igenom två exempel; ett väldigt enkelt, och ett lite mer komplicerat.

### Exempel 1

I mina data om gymnasieskolan skriver jag så här för att slå samman individer som läser på samma skola:

AGGREGATE

/OUTFILE='AGGR.SAV'

/BREAK=gysk\_98

/antal = NU(gysk\_98).

Först har jag skrivit AGGREGATE för att ange att jag skall göra en aggregering. På nästa rad säger jag att den nya tabellen skall sparas i filen AGGR.SAV. Eftersom jag inte skrivit någon ”path”, det vill säga fullständig sökväg till filen, kommer den att sparas i samma bibliotek som mina ursprungsdata ligger i. På min dator hade det fungerat lika bra att skriva ’C:\Sverker\gbg\AGGR.SAV’, eftersom mina ursprungsdata ligger i biblioteket ’C:\Sverker\gbg’.

Raden som inleds med /BREAK säger att alla rader som har samma värde på variabeln gysk\_98 skall slås samman till en observation i den nya tabellen. Detta innebär att om det förekommer 15 olika skolor i ursprungsmaterialet, kommer den nya tabellen att få 15 rader – en för varje skola.

Den sista raden säger att den nya tabellen skall ha en variabel som innehåller antalet individer på varje skola. Först på raden står det /antal, vilket säger att variabeln med antal individer per skola skall heta antal. Efter likhetstecknet står det NU(gysk\_98). Detta betyder att antalet ”cases”, det vill säga antalet rader, som slagits samman skall hamna i den nya variabeln (NU står för Number of cases – Unweighted). Eftersom NU bara räknar antalet rader så spelar det ingen roll vilken variabeln vi använder, så i mitt fall hade det gått lika bra att skriva NU(xgyall). Det som är viktigt är dock att variabeln man använder har värden på alla rader. Om det finns luckor (”system missing”) så räknas inte dessa rader, vilket i och för sig kan vara användbart i vissa fall.

Jag tycker att beskrivningen av de olika sätten att bygga upp ny information som finns i *Syntax Guide* under menyn *Help* i SPSS är riktigt bra. Titta på avsnittet *Aggregate functions* i kapitlet *AGGREGATE*!

### Exempel 2

Vi skall nu göra en lite mer komplicerad övning. Syftet med övningen är rent didaktiskt, och jag vill inbjuda läsaren till reflektion över det sociologiska innehållet i den information vi tar fram. I exemplet kommer vi relatera de sociologiskt intressanta kategorierna Skola, Betyg, Inkomst och Klass till varandra, på ett förmodligen helt nytt sätt. Den första delen av den syntax vi skall arbeta med ser ut så här:

AGGREGATE

/OUTFILE='AGGR2.SAV'

/BREAK=gysk\_98 mbt

/totant = NU(gysk\_98)

/hogrkla = FIN(h90soc3,1,1)

/hogink = FIN(h98sin1,5,5) .

Som tidigare anger vi ett filnamn där den nya tabellen skall sparas. Den första skillnaden gentemot tidigare är att vi anger två olika variabler efter /BREAK. Detta innebär att endast de rader som har exakt samma värden på båda dessa variabler slås samman i den nya tabellen. I detta fall får vi alltså en rad för varje unik kombination av skola och betygskategori.

I exempel 1 bildade vi bara en ny variabel med information i den aggregerade tabellen. Här har vi tre nya variabler. Den första, som helt enkelt räknar antalet individer, har vi behållit sedan tidigare. Här räknar den antalet individer som tillhör en viss betygskategori, och läser på en viss skola. Nästa variabel använder funktionen FIN, som står för Fraction in range. Denna funktion ger oss andelen av de individer som slås samman som har värden på den variabel vi angivit, här h90soc3, inom ett specificerat intervall, här intervallet mellan 1 och 1. Detta intervall innehåller bara värdet 1, men vi skulle kunnat skrivit FIN(h90soc3, 1,3), och skulle då fått andelen av individerna som har värden mellan 1 och 3. Variabeln h90soc3 anger i mina data vilken social klass varje individ tillhör, och värdet 1 står för Högre klass. Vi får alltså andelen individer, på en viss skola, inom en viss betygskategori, vars familj är klassificerad som tillhörande Högre klass.

Den sista raden använder också funktionen FIN, men gör en beräkning med utgångspunkt från variabeln h98sin1. Denna variabel säger vilka inkomstkategori elevens familj tillhör, och värdet 5 står för en inkomst högre än den 9:e decentilen, det vill säga höginkomsttagare.

När vi kör detta script får vi alltså en fil som innehåller:

* Antalet individer på en viss skola som hör till en viss betygskategori.
* Andelen av dessa individer vars familj hänförs till högre klass.
* Andelen av individerna vars familjs inkomst tillhör den 9:e eller 10:e decentilen, det vill säga andelen höginkomsttagare, om vi drar gränsen till höginkomsttagare vid den 9:e decentilen.

I data kan vi nu utläsa att förhållandet mellan Skola, Grundskolebetyg och Inkomst är ganska komplicerat. För lågstatusskolor som Angeredsgymnasiet, Katrinelundsgymnsiet och Burgårdens gymnasium är andelen elever med ursprung i högre klass, likväl som andelen elever med ursprung i familjer med hög inkomst, som störst för elever med måttligt höga betyg, och lägre både för elever med riktigt låga och riktigt höga betyg. Vad kan detta bero på?

För att gå vidare med denna lilla undersökning skapar vi en variabel där andelen elever med ursprung i högre klass relateras till andelen elever med ursprung i en familj med hög inkomst:

IF (hogink > 0) frac = hogrkla / hogink .

EXECUTE .

FORMATS frac (F2.1) .

Observera att vi nu arbetar med filen AGGR2.SAV, det vill säga tabellen där individerna aggregerats på grundval av skola och betygskategori. Den första raden säger att om andelen höginkomsttagare är större än noll, så skall en ny variabel, frac, få ett värde som är förhållandet mellan andelen elever med ursprung i högre klass och andelen elever med ursprung i familjer med hög inkomst. Vi använder IF för att undvika division med noll. För att få med en decimal använder vi FORMATS på den nya variabeln: F2.1 betyder att två siffror skall visas, och av dem skall en siffra stå efter decimalkommat.

Vi gör nu en ny aggregering:

AGGREGATE

/OUTFILE='AGGR3.SAV'

/BREAK=mbt

/hogrkla = FGT(frac,0.5) .

Den nya filen får namnet AGGR3.SAV, och är alltså en aggregering gjort på en tabell som också skapats med hjälp av aggregering. I den nya filen får vi en rad för varje betygskategori, eftersom vi angivit mbt efter /BREAK. Den enda nya variabel vi skapar använder funktionen *FGT – Fraction Greater Than*. Vi får andelen skolor, inom varje betygskategori, där förhållandet mellan andelen elever med ursprung i högre klass och andelen elever med ursprung i familjer med hög inkomst är sådant att andelen elever med ursprung i högre klass är minst dubbelt så stor som andelen elever med ursprung i familjer med hög inkomst.

Med utgångspunkt från den nya filen, AGGR3.SAV, gör vi (bara för att det går) ett stapeldiagram

GRAPH

/BAR(SIMPLE)=VALUE( hogrkla )

/TITLE= 'Föga meningsfull information?'.

som får följande utseende:



Tolkningen av stapeldiagrammet lämnas som övning åt läsaren. Kanske kan Du hitta på ett ännu bättre sätt att använda kommandot AGGREGATE på?

## Transponering av data

I de allra flesta fall när man arbetar med registerdata ligger varje observation (eller individ) på en egen rad, och varje variabel har en egen kolumn. Om det utgångsmaterial man arbetar med har en annan struktur, där till varje variabel har en rad, och varje individ en kolumn, måste man ”transponera” datamatrisen för att kunna arbete med den på vettigt sätt i SPSS. Att transponera innebär just att man gör kolumner av raderna och vice versa. Kommandot man använder för detta heter FLIP, och fungerar fristående. Man kan alltså skriva

FLIP .

för att transponera data. Observera att ett meningsfullt användande av detta kommando förutsätter att antalet rader i filen är relativt litet.

# Automatisk filtrering efter ett antal kriterier

Att skapa en variabel som markerar om en individ läser vid en ”elitutbildning” kräver ett ganska stort antal steg. Dessa har jag satt samman så att de kan utföras i ett svep. Här kommer först hela programmet. Sedan går jag igenom steg för steg vad som händer.

IF (any(**h90sin2**,10)) **xink**=1 .

EXECUTE .

VALUE LABELS **xink** 1 'Hög inkomst'.

IF (**hsun\_n**=6) **xutb**=1 .

IF (**hsun\_n**=7) **xutb**=1 .

EXECUTE .

VALUE LABELS **xutb** 1 'Längre Högskoleutb.' .

IF (**mbt**=6) **xbet**=1 .

IF (**mbt**=7) **xbet**=1 .

EXECUTE .

VALUE LABELS **xbet** 1 'Högt betyg' .

IF (**h90sei2**=1) **xkla**=1 .

EXECUTE .

VALUE LABELS **xkla** 1 'Hög klass' .

COMPUTE **ett** = 1.

EXECUTE .

AGGREGATE OUTFILE='TMP.SAV'

/BREAK = **tgyall**

/**GYF** = NU(**ett**)

/**INKF** = NU(**xink**)

/**UTBF** = NU(**xutb**)

/**BETF** = NU(**xbet**)

/**KLAF** = NU(**xkla**) .

EXECUTE .

AGGREGATE OUTFILE='TMP2.SAV'

/BREAK = **ett**

/**TGY** = NU(**ett**)

/**TINK** = NU(**xink**)

/**TUTB** = NU(**xutb**)

/**TBET** = NU(**xbet**)

/**TKLA** = NU(**xkla**) .

EXECUTE .

SORT CASES BY **tgyall** .

MATCH FILES /FILE=\*

/TABLE='TMP.SAV'

/BY **tgyall**.

EXECUTE.

MATCH FILES /FILE=\*

/TABLE='TMP2.SAV'

/BY **ett**.

EXECUTE.

COMPUTE **xinke** = 0.

COMPUTE **xutbe** = 0.

COMPUTE **xbete** = 0.

COMPUTE **xklae** = 0.

EXECUTE .

IF (**gyf** > 19 & **inkf**/**gyf** > 1.5\***tink**/**tgy** ) **xinke** = 1 .

IF (**gyf** > 19 & **utbf**/**gyf** > 1.5\***tutb**/**tgy** ) **xutbe** = 1 .

IF (**gyf** > 19 & **betf**/**gyf** > 2\***tbet**/**tgy** ) **xbete** = 1 .

IF (**gyf** > 19 & **klaf**/**gyf** > 1.5\***tkla**/**tgy** ) **xklae** = 1 .

EXECUTE .

USE ALL.

FILTER BY **xinke**.

EXECUTE .

FREQUENCIES

VARIABLES=**xgyall**

/ORDER= ANALYSIS .

USE ALL.

FILTER BY **xutbe**.

EXECUTE .

FREQUENCIES

VARIABLES=**xgyall**

/ORDER= ANALYSIS .

USE ALL.

FILTER BY **xbete**.

EXECUTE .

FREQUENCIES

VARIABLES=**xgyall**

/ORDER= ANALYSIS .

USE ALL.

FILTER BY **xklae**.

EXECUTE .

FREQUENCIES

VARIABLES=**xgyall**

/ORDER= ANALYSIS .

COMPUTE **xelit** = 0 .

EXECUTE .

IF (**xinke** = 1 | **xutbe** = 1 | **xbete** = 1 | **xklae** = 1) **xelit** = 1 .

EXECUTE .

USE ALL.

FILTER BY **xelit**.

EXECUTE .

FREQUENCIES

VARIABLES=**xgyall**

/ORDER= ANALYSIS .

## Genomgång av kommandon

Kortfattat gör programmet följande: Ett antal variabler skapas som säger om en viss *individ* tillhör eliten eller inte. Sedan skapas en fil som innehåller information om hur många individer på varje utbildning som uppfyller varje elitkriterium, och en fil som innehåller det totala antalet individer på varje utbildning och det totala antalet individer som uppfyller varje elitkriterium. Denna information förs över till den ursprungliga datafilen.

Sedan skapas ett antal variabler som säger om elever som tillhör eliten är *överrepresenterade* på en viss utbildning. Det är för att kunna mäta förhållandet mellan hur många elitelever det finns på en viss utbildning och hur många det *borde ha funnits*, som vi behövde veta det totala antalet elever dels på varje utbildning. Sedan skapas en filtervariabel som är ett om den utbildning eleven går på uppfyller *något* av elitkriterierna.

Det första som händer är att fyra variabler skapas som säger om en viss *individ* till hör ”eliten”, enligt våra kriterier. Enligt dessa kriterier måste man antingen ha en förälder med inkomst i 10:e decentilen, en förälder med minst 3:årig högskoleutbildning, över 3,9 i grundskolebetyg eller ha en förälder som tillhör högre klass.

IF (any(**h90sin2**,10)) **xink**=1 .

EXECUTE .

VALUE LABELS **xink** 1 'Hög inkomst'.

IF (**hsun\_n**=6) **xutb**=1 .

IF (**hsun\_n**=7) **xutb**=1 .

EXECUTE .

VALUE LABELS **xutb** 1 'Längre Högskoleutb.' .

IF (**mbt**=6) **xbet**=1 .

IF (**mbt**=7) **xbet**=1 .

EXECUTE .

VALUE LABELS **xbet** 1 'Högt betyg' .

IF (**h90sei2**=1) **xkla**=1 .

EXECUTE .

VALUE LABELS **xkla** 1 'Hög klass' .

Vi skapar sedan en variabel som alltid är 1, för att summera totala antalet individer på varje utbildning.

COMPUTE **ett** = 1.

EXECUTE .

Nästa steg är att skapa en ny fil med hjälp av kommandot AGGREGATE. Detta kommando skapar en fil där varje rad motsvarar ett specifikt värde på en viss variabel. Här har jag använt min variabel för *utbildning* (skola+program), eftersom det vi skall välja ut är ett antal *utbildningar* som uppfyller vissa elitkriterier. På detta ställe i koden måste Du ändra så att Din variabel, i textformat, för utbildning används. Kommandot AGGREGATE fungerar så att varje unikt värde i variabeln som är BREAK-variabel får en egen rad i en ny fil vars namn anges som OUTFILE. I denna fil kan man skapa nya variabler vars värden kan räknas ut på olika sätt. Det jag använt är NU(variabel) som innebär att värdena i variabel helt enkelt summeras. ”/GYF” summerar variabeln ett, vilket gör att denna variabel innehåller det totala antalet individer på varje utbildning. ”/INKF” summerar variabeln xink, som är ett om individen har en förälder med hög inkomst. Alltså säger ”/INKF” hur många individer på varje utbildning som uppfyller elitkriteriet för inkomst. På samma sätt fungerar ”/UTBF”, ”/BETF” och ”/KLAF”.

AGGREGATE OUTFILE='TMP.SAV'

/BREAK = **tgyall**

/**GYF** = NU(**ett**)

/**INKF** = NU(**xink**)

/**UTBF** = NU(**xutb**)

/**BETF** = NU(**xbet**)

/**KLAF** = NU(**xkla**) .

EXECUTE .

På samma sätt skapar vi en fil som innehåller det totala antalet individer som uppfyller varje elitkriterium. Här använder vi ett som BREAK-variabel, vilket gör att vi bara får en enda rad i den nya filen. Detta verkar kanske märkligt, men vi behöver det totala antalet individer som uppfyller varje elitkriterium för att kunna säga hur många individer som ”borde” uppfylla varje kriterium på en viss utbildning, om eleverna varit slumpmässigt fördelade.

AGGREGATE OUTFILE='TMP2.SAV'

/BREAK = **ett**

/**TGY** = NU(**ett**)

/**TINK** = NU(**xink**)

/**TUTB** = NU(**xutb**)

/**TBET** = NU(**xbet**)

/**TKLA** = NU(**xkla**) .

EXECUTE .

Vi vill nu lägga till de variabler vi skapat till den ursprungliga datafilen. För att göra detta måste vi först se till att den är sorterad på samma sätt som de nya filer vi skapat. Dessa är sorterade efter utbildning, så därför sorterar vi även vår ursprungsfil efter utbildning.

SORT CASES BY **tgyall** .

Sedan lägger vi till variablerna från de filer vi skapat till vår huvudfil:

MATCH FILES /FILE=\*

/TABLE='TMP.SAV'

/BY **tgyall**.

EXECUTE.

MATCH FILES /FILE=\*

/TABLE='TMP2.SAV'

/BY **ett**.

EXECUTE.

Vi skapar nu ett antal variabler som skall bli filtervariabler för de respektive elitkriterierna. De variabler vi redan har säger om en viss *individ* tillhör eliten. De nya variablerna skall säga om en viss *utbildning* är en elitutbildning. Detta innebär att variabeln skall ha samma värde för alla individer som läser på samma utbildning.

COMPUTE **xinke** = 0.

COMPUTE **xutbe** = 0.

COMPUTE **xbete** = 0.

COMPUTE **xklae** = 0.

EXECUTE .

Vi skapar de nya elitfiltervariablerna med hjälp av två villkor.

1. Antalet individer på utbildningen skall vara minst 20.
2. Individer som tillhör elitkriteriet måste vara överrepresenterade med en faktor minst 2 för betygskriteriet och minst 1,5 för övriga kriterier. Dessa konstanter kan man ändra så att man får lagom många elitutbildningar för varje kriterium.

IF (**gyf** > 19 & **inkf**/**gyf** > 1.5\***tink**/**tgy** ) **xinke** = 1 .

IF (**gyf** > 19 & **utbf**/**gyf** > 1.5\***tutb**/**tgy** ) **xutbe** = 1 .

IF (**gyf** > 19 & **betf**/**gyf** > 2\***tbet**/**tgy** ) **xbete** = 1 .

IF (**gyf** > 19 & **klaf**/**gyf** > 1.5\***tkla**/**tgy** ) **xklae** = 1 .

EXECUTE .

För att vi skall kunna se hur många utbildningar vi fått med från varje kriterium skapar vi frekvenstabeller, där vi filtrerat ut elitutbildningarna för ett kriterium i taget. Om man ser att antalet utbildningar för något kriterium är väldigt litet eller väldigt stort (mindre troligt) kan man ändra konstanterna som säger hur mycket överrepresenterade eliteleverna måste vara på en utbildning för att den skall räknas som elitutbildning.

USE ALL.

FILTER BY **xinke**.

EXECUTE .

FREQUENCIES

VARIABLES=**xgyall**

/ORDER= ANALYSIS .

USE ALL.

FILTER BY **xutbe**.

EXECUTE .

FREQUENCIES

VARIABLES=**xgyall**

/ORDER= ANALYSIS .

USE ALL.

FILTER BY **xbete**.

EXECUTE .

FREQUENCIES

VARIABLES=**xgyall**

/ORDER= ANALYSIS .

USE ALL.

FILTER BY **xklae**.

EXECUTE .

FREQUENCIES

VARIABLES=**xgyall**

/ORDER= ANALYSIS .

Om vi har lagom många utbildningar (ungefär 10 % av totala antalet utbildningar) på varje elitkriterium kan vi gå vidare. Vi skapar den variabel som slutligen skall säga om en utbildning är en elitutbildning. Villkoret skall vara att utbildningen uppfyller *något* av elitkriterierna.

COMPUTE **xelit** = 0 .

EXECUTE .

IF (**xinke** = 1 | **xutbe** = 1 | **xbete** = 1 | **xklae** = 1) **xelit** = 1 .

EXECUTE .

USE ALL.

FILTER BY **xelit**.

EXECUTE .

Slutligen skapar vi en frekvenstabell för att se vilka utbildning vi fick med. Är de för många eller för få kan man gå tillbaka och ändra konstanterna som nu är 1,5 och 2.

FREQUENCIES

VARIABLES=**xgyall**

/ORDER= ANALYSIS .

Slutligen kan man ta bort alla de variabler som skapats under resans gång och inte längre behövs.

# Sammanslagning av flera dataset och skapande av deldataset

Dataregister som beställes från SCB byggs ofta upp utifrån ett antal olika register. Forskningsgruppen för utbildnings- och kultursociologis dataregister över elever i svensk gymnasieskola åren 1997 och 1998 är uppbyggt på följande register:

1. SCB, Register över elever i gymnasieskolan (1997 och 1998) (Bland annat Studievägskod (linje/program/gren/variant), Årskurs, Skolkod (inklusive uppgift om län och kommun), Uppgift om skolans huvudman)
2. SCB, Åk 9-registret 1995–1998 (Skolkod (inklusive uppgift om län och kommun), Uppgift om skolans huvudman, Medelbetyg, Separata betyg (samtliga) Uppgifter om tillval och hemspråk)
3. SCB, Registret över sökande/antagna till gymnasieskolan 1995–1998 (Sökt: Utbildning (1:a, 2:a och 3:e handsval), Sökår, Antagen: Utbildning, Antagen: Skolkod (inklusive uppgift om län och kommun)
4. SCB, Register över avgångna från gymnasieskolan 1998 (Skolkod (inklusive uppgift om län och kommun), Jämförelsetal, Betyg i enskilda ämnen)
5. SCB, RTB (1999) Uppgifter om individen, föräldrarna (fader resp. moder) (civilstånd, medborgarskap, födelseland, invandringsår, inkomst (arbetsinkomst, sammanräknad inkomst), församlingskod)
6. SCB, Barnbanden för FoB 1990, Uppgifter om barnet, föräldrarna (fader resp. moder) och/eller hushållet (ålder, kön, civilstånd, medborgarskap, födelseland, invandringsår, utbildning (SUN-kod), familjetyp, Socioekonomisk indelning (SEI-kod), yrkeskod (NYK) (femställig), näringsgrenskod, sektorstillhörighet, förvärvsarbetets omfattning, yrkesställning, sysselsättning, arbetsplatsens storlek, inkomst (arbetsinkomst, sammanräknad inkomst), hustyp, upplåtelseform, lägenhetstyp (storlek), utrymmesstandard, hushållets sammansättning, hushållstyp, län-, kommun-, församlingskoder)

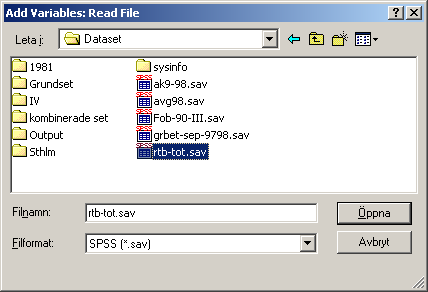
När SCB levererar beställda data får man dessa i form av ett antal separata register. Registren är avidentifierade (personnummer – nödvändiga för att SCB ska kunna hämta uppgifter om en individ från olika register – är ersatta med ett löpnummer) och individbaserad (varje rad i registren motsvaras av en individ). Individerna förekommer i ett flertal register och tack vare löpnumret (som är samma för en enskild individ i varje register) kan man koppla samman de olika registren. Det är dock mycket praktiskt att få data uppdelade på ett antal olika register. Skulle de vara levererade i ett enda register skulle detta bli gigantiskt (innehålla hundra eller tusentals variabler) och bli väldigt svårt att bearbeta. När man dock gjort nödvändiga bearbetningar måste man kunna slå samman de olika registren och detta skall följande avsnitt handla om. Därefter kommer vi in på hur man gör för att dela upp register samt hur man gör om ett register där individerna förekommer på flera rader till ett register där de endast förekommer på en rad.

## Sammanslagning av dataset

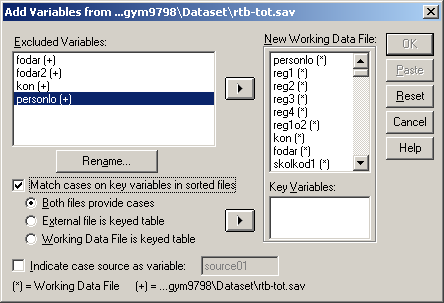
För att slå samman dataset kan man göra på två olika sätt. Om båda dataseten omfattar *samma* variabler väljer man att lägga till rader (*Add Cases*). Om dataseten omfattar *olika* variabler väljer man att lägga till variabler (*Add Variables*). Det senare brukar vara vanligare så låt oss börja där.

### Att lägga till variabler från ett annat dataset

Gå in på *Data/Merge Files/Add Variables...*. Följande ruta uppkommer:



Leta reda på det dataset du vill hämta de nya variablerna från, markera filen och klicka *Öppna*. Du får upp följande fönster:

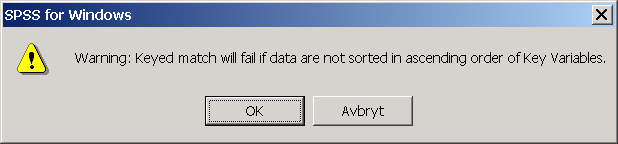


De variabler i det dataset du vill lägga till variabler från som har identiska namn med variabler i ditt ursprungsset exkluderas och hamnar i fönstret *Excluded Variables*. Bland dessa finner vi personlp, individens löpnummer som är identiskt i de dataset individen förekommer. Vi markerar personlp, klickar på *Match cases on key variables in sorted files*, och därefter på *Both files provide cases*. Det vi har gjort är att tala om att vi vill att om personlp har samma värde i de båda filerna så ska individerna hamna på samma rad i den nya datafilen. personlp är Key Variable i detta fall och klickas sedan in i fältet för *Key Variables*. Funktionen *Both files provide cases* betyder att om individer (*cases*) från båda datafilerna kommer att hamna i det nya datasetet.

När vi vill att endast individer från det ursprungliga datasetet ska återfinnas i det nya datasetet men att information om dessa individer skall läggas på från ett annat dataset (som i sin tur kan innehålla andra individer som inte omfattas av det ursprungliga datasetet) väljer vi *External file is keyed tabel*. Detta är en lämplig funktion om vi vill lägga på data om en viss variabel, låt säga variabeln skola. Vi kan då tänka oss att vi har ett register över skolor med en rad variabler som gäller skolorna. I vårt individregister har vi endast information om vilken skola eleverna går på. Om vi utgår från individregistret kan vi lägga på den övriga information om skolorna från skolregistret. Vi markerar då skol-variabeln och inte personlp som *Key Variable*,väljer sedan *External file is keyed tabel* och klickar in skol-variabeln i rutan för *Key Variables*.

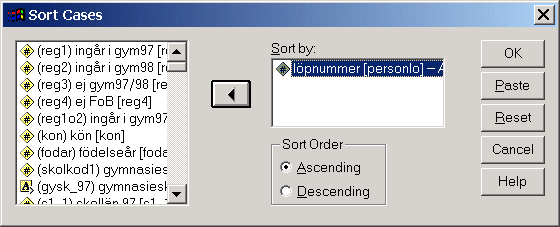
Notera också att variablerna är olika markerade. De variabler som härrör från den ursprungliga filen (*Working file*) är markerade med \* och variabler som läggs till från den andra filen är markerade med +-. Man kan även välja bort variabler från någon av filerna som man ej vill ha med i den nya filen genom att markera den och föra dem till *Excluded variables* med hjälp av piltangenten. Slutligen kan man även döpa om en variabel i *Excluded variables* och föra in den i det nya datasetet.

När vi väl specificerat hur vi ska slå samman registren trycker vi på *OK* eller *Paste*. Det kommer då upp ett varningsfönster:



#### Sortera sina data

Det är viktigt att man har sorterat sina data i stigande skala utifrån den variabel man ämnar slå samman sina dataset utifrån, exempelvis personlp. Detta gör man genom att gå in på *Data/Sort Cases…*. Välj sedan den variabel som man vill sortera utifrån och om sorteringsordningen skall vara stigande eller fallande (välj här stigande).



Syntaxen för detta är enkel:

SORT CASES BY

**personlo** (A) .

#### Syntax för sammanslagning av dataset

*Both files provide cases:*

MATCH FILES /FILE=\*

/FILE='C:\Statistikfiler\gym9798\Dataset\kombinerade set\rtb-tot.sav'

/RENAME (**fodar fodar2 kon** = d0 d1 d2)

/BY **personlo**

/DROP= d0 d1 d2.

EXECUTE.

*External file is keyed tabel:*

MATCH FILES /FILE=\*

/TABLE='C:\Statistikfiler\gym9798\Dataset\kombinerade set\rtb-tot.sav'

/RENAME (**fodar fodar2 kon** = d0 d1 d2)

/BY **personlo**

/DROP= d0 d1 d2.

EXECUTE.

*Working Data File is keyed tabel:*

MATCH FILES /TABLE=\*

/FILE='C:\Statistikfiler\gym9798\Dataset\kombinerade set\rtb-tot.sav'

/RENAME (**fodar fodar2 kon** = d0 d1 d2)

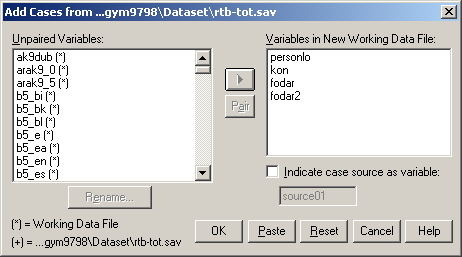
/BY **personlo**

/DROP= d0 d1 d2.

EXECUTE.

### Att lägga till individer från ett annat dataset

Logiken bakom att lägga till *cases* (individer eller observationer) från ett annat dataset skiljer sig helt från den ovan beskrivna för att lägga till variabler. När vi lägger till variabler vill vi att dessa ska ha andra namn än de vi har i ursprungsdatasetet. När vi vill lägga till individer vill vi att variablerna ska ha samma värden som i ursprungsdatasetet. Ifall vi lägger till cases från den datafil vi valde ovan får vi nu endast fyra variabler som har samma namn:



Det är således viktigt att man från början ser till att ha samma namn på variablerna (förutsatt naturligtvis att de innehåller samma typ av information) i de olika dataseten om man sedan har för avsikt att slå samman dem genom att lägga till *cases*. Det är viktigt att strängvariabler har samma *width*, här räcker det inte med att endast ha samma namn.

Följande skript genereras (här kraftigt förkortat):

ADD FILES /FILE=\*

/RENAME (**ak9dub arak9\_0 arak9\_5** […] **termin1 termin2 test tv\_5** = d0 d1 d2 […] d343 d344 d345)

/FILE='D:\Statistikfiler\gym9798\Dataset\rtb-tot.sav'

/RENAME (**aldinv aldinv2 aldinv3** […] **utvnfp\_2 utvnfp\_3 utvnfp\_4** = d346 d347 d348 […] d543 d544 d545)

/DROP= d0 d1 d2 […] d543

d544 d545.

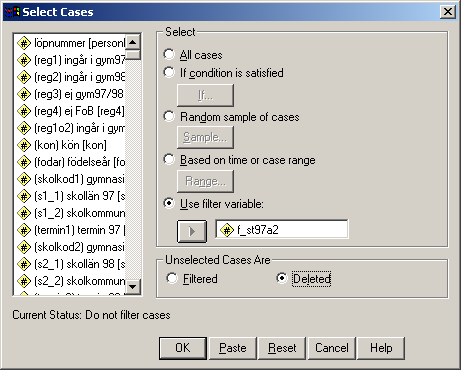
EXECUTE.[[12]](#footnote-12)

Syntaxen för att slå samman dataset där de flesta variablerna inte överensstämmer blir tämligen komplexa och här rekommenderar jag att man skapar ett skript via menyn.

Vill man hålla reda på från vilka dataset de olika observationerna kommer från kan man markera *Indicate case source as variable*. Då skapas en ny variabel som indikerar från vilket dataset variabeln härrör.

## Att skapa deldataset

För att skapa deldataset går vi till väga som när vi filtrerar ett dataset (se kapitel X Filter) med ett undantag. Istället för att använda den för att använda den förinställda *Filtered* (filtrerade) väljer vi *Deleted* (borttagna) under *Unselected Cases Are*.



I och med att de *cases* som inte väljs tas bort är det viktigt att börja med att **spara filen i ett nytt namn** helst innan du filtrerar datasetet. Observera att du då inte har gjort några ändringar sedan du sparade datasetet under det ursprungliga namnet (dessa förändringar kommer annars att försvinna för de cases som du filtrerar bort). Om du inte sparar datasetet innan du filtrerar det är det viktigt att du därefter går in på *File/Save as* och sparar filen under ett nytt namn. Använd **inte** kortkommandot <ctrl+s> då sparas filen i det ursprungliga namnet och alla cases som du tagit bort försvinner för gott (såvida du inte har en tidigare backup av ursprungsfilen).

Skriptet för att välja ut cases och ta bort cases som inte är valda är

FILTER OFF.

USE ALL.

SELECT IF(NOT(**f\_st97a2** =0)).

EXECUTE .

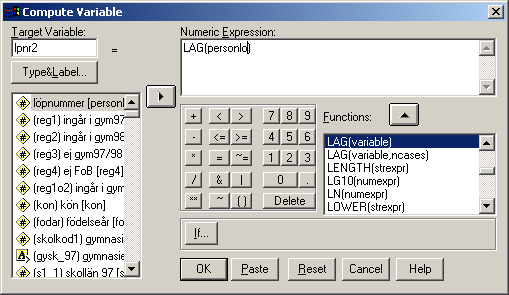
## Att skapa dataset där individerna utgör endast en rad

Det händer att data levereras från SCB på så sätt att en individ kan förekomma på fler rader. Detta är fallet med registrerade högskoleutbildningar. En individ registreras i SCBs register på en kurs. För en och samma termin kan en individ vara registrerad på 10-tals olika kurser (ibland upp till 30-talet kurser). Av utrymmesskäl så är det då enklast att leverera dessa data på så sätt att individerna förekommer på flera rader, dvs. att varje enskild kurs blir en rad eller observation. Man slipper då få ett dataset med en massa tomma celler.[[13]](#footnote-13) Men vid bearbetningen av individerna är det ofta nödvändigt att individerna endast förekommer på en rad och därmed endast utgör en observation. Detta går att åstadkomma genom att skapa ett antal deldataset och sedan slå samman dem till ett nytt dataset.

Det första vi gör är att skapa en variabel som förskjuter löpnumren ett steg. Principen är följande:

|  |  |  |
| --- | --- | --- |
| Lpnr | lpnr2 | dub |
| 1 | , | , |
| 2 | 1 | , |
| 4 | 2 | , |
| 4 | 4 | 1 |
| 7 | 4 | , |
| 9 | 7 | , |
| 9 | 9 | 1 |

För att skapa en variabel som förskjuter löpnumret ett steg använder vi funktionen *Lag*. Gå in på *Transform/Compute* och välj funktionen *Lag*. Välj variabeln med löpnummer, här döpt till personlo. Skriv in en ny Target Variable, exempelvis lpnr2. Tryck på *Paste*.



Följande skript erhålls.

COMPUTE **lpnr2** = LAG(**personlo**) .

EXECUTE .

Det vi därefter vill skapa är en variabel som talar om ifall variablerna personlp och lpnr2 har samma värde. Detta görs genom att omkoda variabeln personlp till en ny variabel, dub, på följande sätt:

DO IF (**personlo** = **lpnr2**) .

RECODE

**personlo**

(1 thru 1000000000 = 1) INTO **dub** .

END IF .

EXECUTE .

Skriptet säger att om personlo är lika med lpnr2 så ska alla värden mellan 1 och 10000000 (dvs. alla värden) kodas om till värdet 1 i variabeln dub.

Ett alternativt och enklare sätt att skriva detta är:

COMPUTE dub=( **personlo** = LAG(**personlo**)).

EXECUTE .

Vi har nu erhållit en struktur av variabler som visas i tabellen ovan. Nu är det möjligt att filtrera ut de individer som är dubblerade och spara dessa i ett nytt dataset. Nedan visas ett exempel från en fil där jag delat upp alla högskoleregistrerade höstterminen 1993 (förklaringar skrivs efter \*-tecken):

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Runda 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.

COMPUTE **dub**=(**lpnr** = LAG(**lpnr**)).

VARIABLE LABEL dub 'lpnr = LAG(**lpnr**) (FILTER)'.

FORMAT **dub** (f1.0).

EXECUTE .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Skapar en ny variabel dub som får etiketten ” dub 'lpnr = LAG(lpnr) (FILTER)' ” med formatet 1.0 (1 tecken och 0 decimaler). Variabeln Dub får värdet 1 ifall löpnumret är dubblerat.

SAVE OUTFILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_ind.sav'

/COMPRESSED.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sparar en ny datafil h93h\_ejinv\_tot\_2\_lprog\_ind.sav.

FILTER OFF.

USE ALL.

SELECT IF(**dub**=0 OR MISSING(**dub**)).

EXECUTE .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Filtrerar ut de individer (*cases*) som **inte har** dubbla löpnummer, dvs. alla individerna återfinns endast en gång i denna nya fil.

SAVE OUTFILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_it.sav'

/COMPRESSED.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sparar en ny datafil h93h\_ejinv\_tot\_2\_lprog\_it.sav.

GET

FILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_ind.sav' .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Öppnar datafil h93h\_ejinv\_tot\_2\_lprog\_ind.sav (där alla individer finns även de som har dubblerade löpnummer).

FILTER OFF.

USE ALL.

SELECT IF(**dub**=1).

EXECUTE .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Filtrerar ut de individer (*cases*) som **har** dubbla löpnummer, dvs. individerna kan återfinnas flera gånger i denna nya fil.

SAVE OUTFILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_i\_1.sav'

/COMPRESSED.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sparar en ny datafil h93h\_ejinv\_tot\_2\_lprog\_i\_1.sav.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Runda 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.

COMPUTE **dub**=(**lpnr** = LAG(**lpnr**)).

VARIABLE LABEL dub 'lpnr = LAG(lpnr) (FILTER)'.

FORMAT **dub** (f1.0).

EXECUTE .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Beräknar variabeln dub som får värdet 1 ifall löpnumret är dubblerat.

SAVE OUTFILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_i\_1.sav'

/COMPRESSED.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sparar datafilen h93h\_ejinv\_tot\_2\_lprog\_i\_1.sav igen, nu med omräkning av variabeln dub.

FILTER OFF.

USE ALL.

SELECT IF(**dub**=1).

EXECUTE .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Filtrerar ut de individer (*cases*) som **har** dubbla löpnummer, dvs. individerna kan återfinnas flera gånger i denna fil.

SAVE OUTFILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_i\_2.sav'

/COMPRESSED.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sparar en ny datafil h93h\_ejinv\_tot\_2\_lprog\_i\_2.sav.

GET

FILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_i\_1.sav' .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Öppnar datafil h93h\_ejinv\_tot\_2\_lprog\_i\_1.sav.

FILTER OFF.

USE ALL.

SELECT IF(**dub**=0 OR MISSING(**dub**)).

EXECUTE .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Filtrerar ut de individer (*cases*) som **inte har** dubbla löpnummer, dvs. alla individerna återfinns endast en gång i denna nya fil.

SAVE OUTFILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_i\_1.sav'

/COMPRESSED.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sparar datafilen h93h\_ejinv\_tot\_2\_lprog\_i\_1.sav igen, nu med individerna endast förekommande en gång i filen.

GET

FILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_i\_2.sav' .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Öppnar datafil h93h\_ejinv\_tot\_2\_lprog\_i\_2.sav.

Därefter upprepar man det hela till man fått fram alla nya filer. Öppna sedan datafilen h93h\_ejinv\_tot\_2\_lprog\_i\_2.sav. Vi ska nu slå samman de olika deldataseten vi skapat. Observera att vi måste döpa om variablerna innan vi slår ihop dataseten! Detta för att kunna lägga till nya variabler.

MATCH FILES /FILE=\*

/FILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93'+

'-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_i\_1.sav'

/RENAME (**dub** = d0) **hv\_lprin**=**hv\_lpi1 k\_id**=**k\_id\_1 ktxp**=**ktxp\_1**

/BY **lpnr**

/DROP= d0.

EXECUTE.

SAVE OUTFILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_it.sav'

/COMPRESSED.

MATCH FILES /FILE=\*

/FILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93'+

'-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_i\_2.sav'

/RENAME (**dub** = d0) **hv\_lprin**=**hv\_lpi2 k\_id**=**k\_id\_2 ktxp**=**ktxp\_2**

/BY **lpnr**

/DROP= d0.

EXECUTE.

SAVE OUTFILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_it.sav'

/COMPRESSED.

MATCH FILES /FILE=\*

/FILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93'+

'-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_i\_3.sav'

/RENAME (**dub** = d0) **hv\_lprin**=**hv\_lpi3 k\_id**=**k\_id\_3 ktxp**=**ktxp\_3**

/BY **lpnr**

/DROP= d0.

EXECUTE.

SAVE OUTFILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_it.sav'

/COMPRESSED.

MATCH FILES /FILE=\*

/FILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93'+

'-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_i\_4.sav'

/RENAME (**dub** = d0) **hv\_lprin**=**hv\_lpi4 k\_id**=**k\_id\_4 ktxp**=**ktxp\_4**

/BY **lpnr**

/DROP= d0.

EXECUTE.

SAVE OUTFILE='D:\Statistikfiler\hgsk93-99\Datafiler\SPSS-filer\h93-99\ejinvfiler\h93h\_ejinv\_tot\_2\_lprog\_it.sav'

/COMPRESSED.

Etcetera tills vi har lagt till alla nya deldataset. Vad vi har gjort kan illustreras av följande förenklade dataset:

Ursprungligt dataset:

|  |  |
| --- | --- |
| lpnr | var1 |
| 1 | v1 |
| 2 | v2 |
| 4 | v3 |
| 4 | v4 |
| 7 | v5 |
| 9 | v6 |
| 9 | v7 |
| 9 | v8 |
| 9 | v9 |
| 13 | v10 |
| 13 | v11 |
| 13 | v12 |
| 18 | v13 |
| 18 | v14 |

Dataset 1 (innehåller alla individer men endast en gång, observera att det är individens första observation som återfinns i detta dataset):

|  |  |
| --- | --- |
| lpnr | var1 |
| 1 | v1 |
| 2 | v2 |
| 4 | v3 |
| 7 | v5 |
| 9 | v6 |
| 13 | v10 |
| 18 | v13 |

Dataset 2 (innehåller alla individer som har minst två observationer; individerna återfinns endast en gång):

|  |  |
| --- | --- |
| lpnr | var1 |
| 4 | v4 |
| 9 | v7 |
| 13 | v11 |
| 18 | v14 |

Dataset 3 (innehåller alla individer som har minst tre observationer; individerna återfinns endast en gång):

|  |  |
| --- | --- |
| Lpnr | var1 |
| 9 | v8 |
| 13 | v12 |

Dataset 4 (innehåller alla individer som har minst fyra observationer; individerna återfinns endast en gång):

|  |  |
| --- | --- |
| Lpnr | var1 |
| 9 | v9 |

När vi sedan slår samman dataseten (och döper om variablerna) erhåller vi följande dataset:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lpnr | var1 | var2 | var3 | var4 |
| 1 | v1 |  |  |  |
| 2 | v2 |  |  |  |
| 4 | v3 | v4 |  |  |
| 7 | v5 |  |  |  |
| 9 | v6 | v7 | v8 | v9 |
| 13 | v10 | v11 | v12 |  |
| 18 | v13 | v14 |  |  |

Resultatet är ett dataset där individerna endast förekommer en gång men där samma typ av information om individerna återfinns i flera variabler. Antalet tomma celler är också betydligt större än i det ursprungliga datasetet.

# Att läsa in textfiler (och därmed skapa nya dataset)

Datafiler som beställs av SCB levereras som textfiler. Dessa kan vara av två typer. Antingen är de rena textfiler där positionerna är avgörande för hur man skapar variabler, eller så är de uppdelade i variabler utifrån någon form av separator såsom tabbtecken. En textfil kan se ut på följande sätt (textfiler kan läsa av Word Anteckningar genom att man dubbelklickar på textfilen):

PERSONLO SKOLKOD STVKOD TERMIN

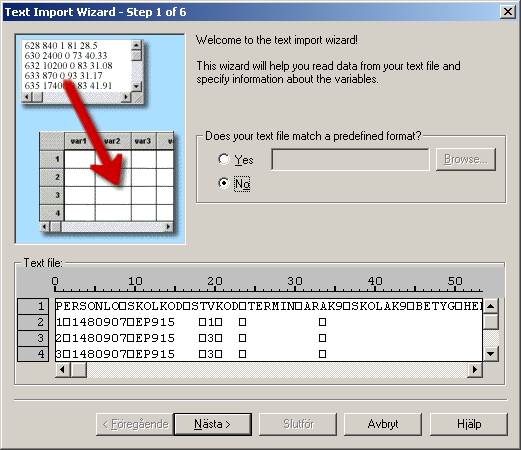
1 1480907 EP915 1

2 1480907 EP915 3

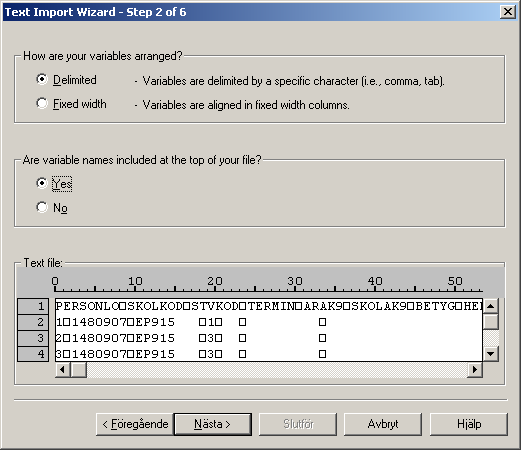
3 1480907 EP915 3

Detta är de fyra första raderna på den textfil som innehåller information om alla elever i gymnasieskolan 1997. Totalt innehåller filen 312933 rader, vilka motsvara 312933 individer, elever i det här fallet, samt en inledande rad som talar om vad variablerna heter. Vi kan utgå från denna fil för att exemplifiera hur det går till att läsa in en textfil i SPSS.

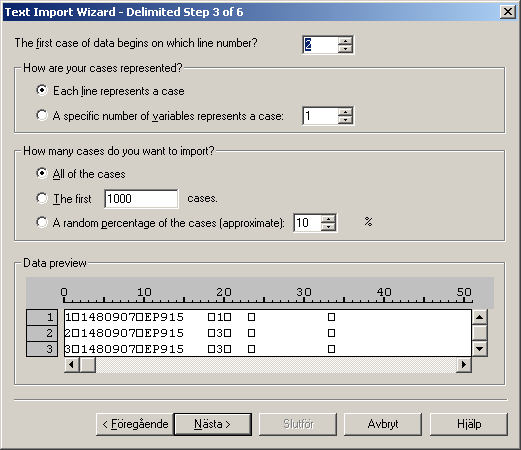
Gå in på *File/Read Text Data....* Markera den fil du vill läsa in och klicka *OK*. Följande ruta uppkommer. Klicka *Nästa*.

****

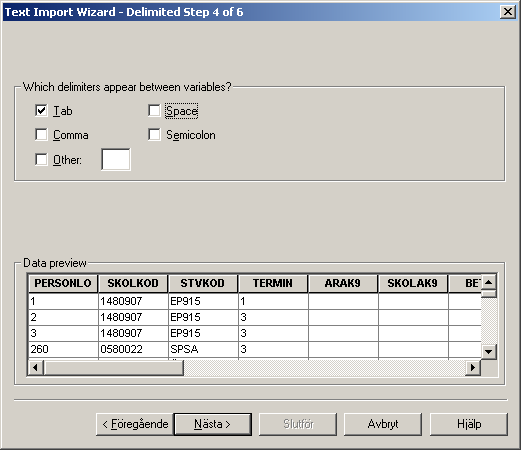
Välj i nästa ruta *Delimited* (som vi ser i *Text File-fönstret* är texten uppdelad med ٱ, vilket innebär att den är separerad med tabbar). Markera sedan att variabelnamne finns överst i filen. Klicka *Nästa*.



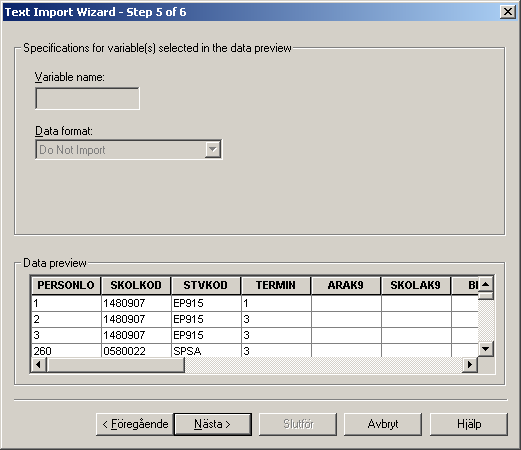
Kontrollera att textfilen börjar importeras på rätt rad. I vårt fall skall det stå rad nummer 2. Välj därefter att importera alla observationer (*Cases*). Notera att en observation inte behöver vara liktydigt med en rad, man kan ha observationer som sträcker sig över flera rader. Klicka *Nästa*.



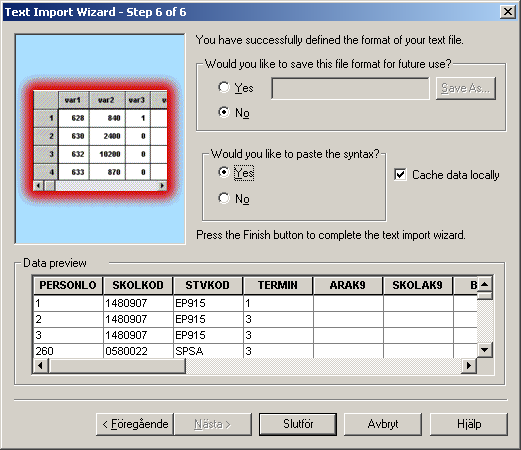
I nästa steg väljer du vilka typer av separatorer som finns i textfilen. I vårt fall skall endast *Tab* kryssas i.



Nästa steg möjliggör specificering av variablerna. Vill vi exempelvis ändra namnet på någon variabel gör vi det i rutan för *Variable name*. Vi kan också bestämma vilken typ av data det är, exempelvis en numerisk variabel eller en textvariabel, genom att ändra i *Data format*. Det går även att välja att inte importera en variabel. Klicka *Nästa* efter att alla inställningar är gjorda.



Markera här *paste syntax*. Det är alltid bra att spara inläsningsskriptet. Dels kanske man behöver läsa in filen igen, dels kanske man vill korrigera något i inläsningen av data.



Det skript vi erhåller ser ut som följer:

GET DATA /TYPE = TXT

/FILE = 'D:\Statistikfiler\gym9798\GymnSCB\Dataregister\Sasdata\pop97.txt'

/DELCASE = LINE

/DELIMITERS = "\t"

/ARRANGEMENT = DELIMITED

/FIRSTCASE = 2

/IMPORTCASE = ALL

/VARIABLES =

PERSONLO F3.2

SKOLKOD F7.2

STVKOD A8

TERMIN F1.0

ARAK9 F2.1

SKOLAK9 F9.2

BETYG F27.2

HEMSP F3.2

TILLVAL F1.0

MEDELBET F2.1

ARSOKT F2.1

U\_ANS\_U1 F6.2

U\_ANS\_U2 F6.2

U\_ANS\_U3 F6.2

U\_UTBSTV F6.2

U\_UTBSKO F7.2

KON F1.0

FODAR F2.1

.

CACHE.

EXECUTE.

Vi kan här göra vissa korrigeringar. För det första Kan vi se att vissa variabler får värden med F framför. Detta betyder att det handlar om numeriska värden. F2.1 betyder att variabel får 2 siffror med 1 decimal. Man kan här justera så att man slipper decimalerna genom att skriva F2.0 eller F2. Variabler med A framför är strängvariabler eller textvariabler.

Vi kan justera alla numeriska variabler så att de inte får decimaler, detta eftersom de inte finns några decimalvärden i ursprungsdata. En annan korrigering är att vi ändrar antalet tecken i variabeln PERSONLO från 3 till 7 (SPSS utgår från de översta raderna i textfilen och där finns inte fler än 3 siffror i PERSONLO, vilket gör att endast 3 tecken föreslås. Man bör här vara observant och kolla med en rad i slutet av textfilen så att det stämmer. Det justerade skriptet ser ut enligt följande:

GET DATA /TYPE = TXT

/FILE = 'D:\Statistikfiler\gym9798\GymnSCB\Dataregister\Sasdata\pop97.txt'

/DELCASE = LINE

/DELIMITERS = "\t"

/ARRANGEMENT = DELIMITED

/FIRSTCASE = 2

/IMPORTCASE = ALL

/VARIABLES =

PERSONLO F7

SKOLKOD F7

STVKOD A8

TERMIN F1

ARAK9 F2

SKOLAK9 F9

BETYG A27

HEMSP F3

TILLVAL F1

MEDELBET F2

ARSOKT F2

U\_ANS\_U1 F6

U\_ANS\_U2 F6

U\_ANS\_U3 F6

U\_UTBSTV F6

U\_UTBSKO F7

KON F1

FODAR F2

.

CACHE.

EXECUTE.

Vi har nu skapat ett nytt dataset. Spara detta. Det är nu viktigt att granska resultaten. Gör ett antal frekvenstabeller så att du ser att värden i variablerna är rimliga. Observera att du bör undvika att ta med löpnummer-variabeln eftersom dessa gör att programmet lätt kraschar pga. det enormt stora antalet unika värden som variabeln omfattar.

Vi ser vidare att variabeln betyg omfattar 27 tecken. Här fattas tabbtecken som separerar de enskilda betygen. För att separera betygen får vi göra en ny inläsning. Vi börjar med att skapa en ny fil som bara innehåller betygsvariabeln. Första steget är att spara datasetet i ett annat namn. Tag sedan bort alla variabler förutom betyg. Spara i det nya namnet. Spara sedan filen som en textfil. Välj *Save As* och ändra Fileformat till Fixed ASCII. Tryck *Paste*. Följande skript erhålls.

WRITE

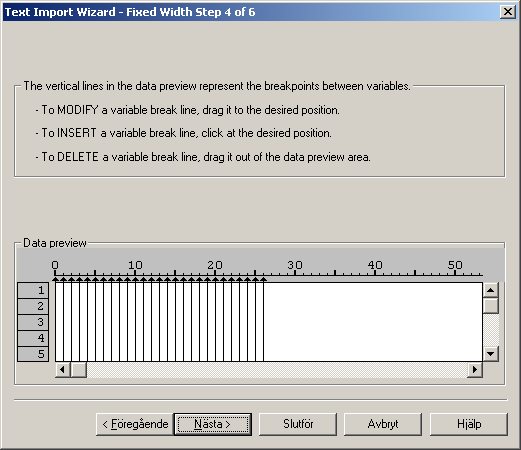
OUTFILE='D:\Statistikfiler\gym9798\GymnSCB\Dataregister\Sasdata\bet.dat'

TABLE /ALL.

EXECUTE.

Gör sedan om proceduren *File/Read Text Data...* Skillnaden mot ovan är att filen denna gång har tillägget ”.dat” och inte ”txt”. När man öppnar filen letar man således efter en fil med filformatet Data (\*.dat). I steg 2 väljer man *Fixed Width* och Ej angivna variabelnamn överst i filen. I steg 3 väljs att importera data från rad 1.

I steg 4 markerar man manuellt med pilarna hur man vill skapa variabelgränserna i textmaterialet. I vårt fall är det 27 variabler med ett tecken vardera som skall skapas. Det markeras med en pil för varje variabel.



I steg 5 kan man skriva in namnen på de nya variablerna. Man kan också välja typ av variabler. Här bör man ställa in att alla variabler är strängvariabler (vissa av betygsvärdena är bokstäver). Ifall defaultvärdet är numeriska värden kan man ändra alla variabler till strängvariabler i skriptet som genereras genom att ersätta ”F1.0” med ”A1”. Det skript som slutligen genereras ser ut som följer:

GET DATA /TYPE = TXT

/FILE = 'D:\Statistikfiler\gym9798\GymnSCB\Dataregister\Sasdata\bet.dat'

/FIXCASE = 1

/ARRANGEMENT = FIXED

/FIRSTCASE = 1

/IMPORTCASE = ALL

/VARIABLES =

/1 V1 0-0 A1

V2 1-1 A1

V3 2-2 A1

V4 3-3 A1

V5 4-4 A1

V6 5-5 A1

V7 6-6 A1

V8 7-7 A1

V9 8-8 A1

V10 9-9 A1

V11 10-10 A1

V12 11-11 A1

V13 12-12 A1

V14 13-13 A1

V15 14-14 A1

V16 15-15 A1

V17 16-16 A1

V18 17-17 A1

V19 18-18 A1

V20 19-19 A1

V21 20-20 A1

V22 21-21 A1

V23 22-22 A1

V24 23-23 A1

V25 24-24 A1

V26 25-25 A1

V27 26-26 A1

.

CACHE.

EXECUTE.

# Att koda data för överföring till SPAD (program för bland annat korrespondensanalys)

Det går relativt enkelt att i SPAD använda ett dataset från SPSS men det krävs dock att data är kodade på ett visst sätt. Följande gäller.

I SPAD finns två typer av variabler kvalitativa (*nominales*) och kvantitativa (*continues*). Variabler som innehåller värden från 1 till och med 300 kodas automatiskt om till kvalitativa variabler när man konverterar en fil från SPSS till SPAD. Övriga variabler kodas automatiskt som kvantitativa. Så fort en variabel innehåller värde 0 blir variabeln kvantitativ. För att göra korrespondensanalyser vill man ha kvalitativa variabler. Därför är det viktigt att iakttaga följande:

1. **Strängvariabler ska göras om till numeriska variabler** med värden från 1 och uppåt, men max med värdet 300 (se i första hand IX.2.8 Automatisk omkodning av strängvariabel till numerisk variabel, men även avsnittet XII.3 Att omkoda strängvariabler till numeriska variabler och använda värden i strängvariabeln till etiketter i den numeriska )
2. **Numeriska variabler med värdet 0 kodas om** så att 0 blir ett värde mellan 1 och 300 (se avsnittet IX.1 Omkodning av samma variabel... och IX.2.11 Några tips om omkodning och konstruktion av ett dataset)
3. **Numeriska variabler kodas så att det inte blir några luckor**, dvs. koda om en variabel med värdena 1, 10 och 99 till värdena 1, 2 och 3. Detta för att SPAD automatiskt lägger in värden (värdena kallas i SPAD *modalité* eller modalitet) som fyller ut upp till det högsta värdet – i det här fallet upp till värdet 98 (se avsnittet IX.1 Omkodning av samma variabel... alternativt avsnitt IX.2.9 Automatisk koncentration värdena i en numerisk variabel)
4. **Skapa identifikationsvariabel.** Ifall man har ett dataset med personer eller institutioner där det är viktigt att kunna särskilja de olika individerna kan man skapa en identifikatör. Detta görs på så sätt man den första variabeln i datasetet görs till en strängvariabel med namnet ”id\_spadn” där man skriver i text namnen på individerna, institutioner eller observationerna. Om man redan har en dylik variabel kopiera denna och lägg den första samt döp den till ”id\_spadn”.
5. **Tänk på att ha ett så pass komplett dataset som möjligt** när du väl konverterar filen till en SPAD-fil. Det är nämligen rätt omständligt att koda om i SPAD. Det går förvisso att göra om SPSS-filen och importera den igen i SPAD, men detta bör undvikas. (Se avsnitt IX.2.12 Några tips för att koda om SPSS-data för att passa korrespondensanalyser i SPAD)
6. **Etiketter max 20 tecken.** Fördelen med SPAD är att det går att dirket importera filer från SPSS och därmed behålla etiketter på variabelvärden. SPAD använder dock endast de första 20 tecknen. Dessutom blir graferna rätt röriga och svårlästa om man har för långa etiketter. Skapa därför så korta etiketter som möjligt, men se till att de fortfarande är begripliga**.** Man kan också tänka på att göra alla etiketter enligt en genomtänkt logik så att alla variabler som rör fadern startar med ”F-”, allasom rör modern startar med ”M-” och alla för hushållet med ”H-”.
7. **Unika etiketter.** Korrespondensanalysen är framför allt lämplig för att analysera ett stort antal variabler. De grafer som programmet genererar blir dock lätt röriga när man använder ett stort antal variabler och variabelvärden. För att kunna tolka graferna är det ytterst viktigt att man skapar etiketter som är unika. Låt säga att vi har ett antal variabler om elevernas betyg, där vi har värdena MVG, VG, G, IG, Ej betyg och Ej läst. Om vi kodar alla betygsvariabler med dessa etiketter kommer vi inte att kunna skilja ett MVG i engelska från ett i tyska. Sätt därför etiketter som ”Gy-bet-Eng-MVG”, ”Gy-bet-Eng-VG”, ”Gy-bet-Eng-G”, etc. och ”Gy-bet-Tys-MVG”, ”Gy-bet-Tys-VG”, ”Gy-bet-Tys-G”, etc. För grundskolebetyg kan man använda prefixet ”Gr-bet” för att på så sätt kunna skilja grundskolebetygen från gymansiebetygen.
8. **Missing values kodas till specifikt värde**. Koda om alla *missing values* till specifikt värde, förslagsvis det högsta värden av variabelvärdena + 1. Undvik framför allt värdet 0 (om man ger *missing values* värdet 0 skapar SPAD en kontinuerlig variabel och inte en nominal). I fall man inte ger *missing values* ett specifikt värde skapar SPAD egna *missing values*, *Reponse manquante*, som ej går att skilja från varandra. Samma princip gäller således här som ovan, gör etiketterna unika, dvs. skriv ”F-yrke-Ej i FoB”, ”M-yrke-Ej i FoB” och ”H-yrke-Ej i FoB” så att det går att skilja de olika *missing values* från varandra.

1. Det finns naturligtvis en risk att nybörjaren finner det hela för komplicerat och den vane användare det mesta för trivialt. Jag hoppas dock att det går att ha överseende med detta och ta till sig det man själv har användning av. [↑](#footnote-ref-1)
2. Karin Dahmström, *Från datainsamling till rapport. Att göra en statistisk undersökning*, Studentlitteratur, Lund 2000. [↑](#footnote-ref-2)
3. Nominal anger att värdena inte kan rangordnas (ex. bostadsort), ordinal att värdena kan rangordnas men att stegen mellan de olika värdena inte är enhetliga (rangordning av motiv för att välja en viss skola) och skalvariabler att värdena kan rangordnas och att stegen mellan värdena är enhetliga (ex. temperatur). [↑](#footnote-ref-3)
4. Ibland kan det dock vara en poäng att dra ut frekvenstabeller för löpnummer, exempelvis om man vill se efter så att inte en och samma individ förekommer mer än en gång. [↑](#footnote-ref-4)
5. Eller mer vardagligt uttryckt vid vilket procenttal. [↑](#footnote-ref-5)
6. Notera att skillnaderna i kronor räknat mellan de olika grupperna kommer att variera. Ett alternativt indelningssätt är att utgå från inkomsterna och dela in populationen i efter inkomstnivå i låt säga 50.000 kr-intervaller. [↑](#footnote-ref-6)
7. Se vidare Karin Dahmström, *Från datainsamling till rapport. Att göra en statistisk undersökning*, Studentlitteratur, Lund 2000, p 29. [↑](#footnote-ref-7)
8. Se vidare, ibid., pp. 151-186. [↑](#footnote-ref-8)
9. Tabellen är skapad på så sätt att den kopierats från SPSS Output till Excel och därefter kopierad in i Word och redigerad i Word (bland annat med Table/TabelAutoFormat). Det går också att klippa in tabellen direkt från SPSS Output till Word, men det är enklare att först göra redigering av siffror i Excel för att sedan kopiera in tabellen i Word. [↑](#footnote-ref-9)
10. Tabellen är skapad på så sätt att den kopierats från SPSS Output direkt till Word. Markera tabellen i SPSS Output, högerklicka och välj *Copy Objects* och klistra sedan in i Word. [↑](#footnote-ref-10)
11. I den engelska versionen av Word, kan vara annorlunda i den svenska. [↑](#footnote-ref-11)
12. För att köra ett skript av denna typ måste man ta bort alla hakparenteser som här satts in för att förkorta skriptet. [↑](#footnote-ref-12)
13. Ett dataset med 10 variabler för kurser och 100 000 observationer ger 1 000 000 celler. Låt säga att detta motsvarar 80 000 faktiska individer (antalet individer blir lägre eftersom vissa individer är registrerade på flera kurser, dvs. observationerna i detta fall). Om det högsta antalet kurser en individ är registrerad på är 20 och vi vill göra om datasetet till ett där varje individ utgör en observation får vi således ett dataset med 10 x 20 x 80 000 celler, dvs. totalt 16 000 000, där de flest cellerna är tomma. [↑](#footnote-ref-13)