Særtryk fra Institut for Økonomi, Politik og Forvaltning, Aalborg Universitet

Indledende statistisk analyse i SAS®

En praktisk vejledning

Henrik Lolle

Indledende statistisk analyse i SAS®
En praktisk vejledning

af Henrik Lolle

Institut for Økonomi, Politik og Forvaltning
Aalborg Universitet
Fibigerstræde 1
9220 Aalborg Ø

tlf. 96358200

ISBN 87-90789-19-9
ISSN 1396-352X
2000:2

tryk: UNI.PRINT, Aalborg Universitet

INDHOLD

FORORD

SAS er et stort, integreret software-system, der - som det af SAS Institute selv formuleres - tilbyder komplet kontrol over data-tilgang, -behandling, -analyse og -præsentation. Der skal i denne lille vejledning løftes en lille flig af sløret til dette system.

Brugervejledningen omhandler hovedsageligt det rent praktiske arbejde, der foregår i forbindelse med *analysearbejde*, efter at der er indhentet data, og der er kun medtaget forholdsvis simple analysemåder. For yderligere specifikation og udførlig vejledning kan henvises til diverse SAS-manualer[1] (f.eks. SAS® Language: Reference; SAS® Procedures Guide; SAS/STAT® User's Guide, Vol.1+2) samt til Andersen m.fl (1995), Spector (1993) og Dilorio (1996). For vejledning i spørgeskemakonstruktion og oprettelse af eget datasæt kan henvises til Nielsen (1998). Med hensyn til spørgeskemakonstruktionen kan også anbefales Olsen (1998), som ser specifikt på måleproblemer i surveyanalyser, og de Vaus (1996), som er en mere generel metodebog til surveyanalyse. Og for vejledning og gode råd vedrørende indhentning af datamateriale til sekundær analyse kan henvises Bentzen m.fl. (1999). Til sidst angående den mindre teknisk-praktisk betonede del af analysefasen kan anbefales bl.a. Hellevik (1994), Frankfort-Nachmias (1997) samt Rosenberg (1968).

Der er selvfølgelig allerede skrevet mange og lange manualer og brugervejledninger til statistisk analyse med softwarepakken SAS, men dels indeholder disse som oftest et hav af informationer, som de fleste brugere aldrig vil få behov for, dels har man som ny bruger af en softwarepakke et stort behov for at få detaljeret vejledning til de første skridt - gerne med konkrete eksempler, som kan bruges som udgangspunkt i ens eget arbejde. Givet vis vil man ret hurtigt få brug for funktioner eller programkode, som ikke bliver beskrevet i det følgende, men når man først føler sig lidt hjemme i SAS-layoutet og forstår logikken i funktionerne og programmeringen, så er det forholdsvist enkelt at sætte sig ind i nye ting ved at bladre lidt rundt i en manual eller bare prøve sig frem på skærmen.

I alle vejledningens eksempler benyttes et øve-datasæt, hvor hver enkelt observation svarer til en skoleelev. Til hver elev er knyttet oplysninger om alder, højde og

[1] SAS-manualer kan være vanskeligt tilgængelige med hensyn til det statistiske stof. Ofte kan det faktisk være en stor hjælp også at kigge lidt i en SPSS-manual, som er lettere tilgængelig. Til gengæld er SAS mere grundig med at få alle detaljer med og især har de langt bedre indeksering end SPSS.

vægt samt elevens svar på en række tilfredshedsspørgsmål angående forskellige aspekter vedrørende skolen. Det er tilstræbt, at vejledningsteksten så at sige har et naturligt forløb, så den simulerer en analyseproces. Dette har den store fordel, at det letter forståelsen for, hvordan en kvantitativ analyseproces kan se ud, og hvad man gør, i hvilken rækkefølge. Det har dog også nogle ulemper. Det er f.eks. vanskeligt at opstille et eksempel på én analyse, hvor samtlige de forskellige data-manipulationer og analyseformer, som man ønsker at gennemgå, naturligt kan passes ind. Af den grund er der da også indføjet enkelte ekskurser, og der er tilføjet et appendiks, hvor der rådes bod på nogle 'hængepartier', hvad angår håndtering af data. Det skal dog her understreges, at samtlige de gennemgåede analyseteknikker er forholdsvis simple, og i visse af vejledningens eksempler ville der kunne benyttes bedre, men også teknisk set mere vanskelige analyseteknikker.

En anden ulempe ved så vidt muligt at simulere en naturlig analyseproces med forklarende tekst, er at det i forhold til deciderede manualer er vanskeligt at gøre teksten nem at bruge som opslagsværk. For at tilpasse sig det konkrete eksempel mister gennemgangen visse steder noget i systematik. Dog er teksten kun på godt 100 sider, hvilket gør den forholdsvis hurtig at læse igennem i ét stræk. Endvidere er der til slut i vejledningen indføjet et indeks med afsnitsangivelser over mange detaljer.

Til slut i denne forbindelse skal det blot nævnes, at man som SAS-bruger efterhånden får sine egne vaner for, hvordan man helt konkret løser problemer både med peg-og-klik og med programskrivning. En række steder anviser jeg forskellige måder, hvorpå samme problem kan løses, men stadigvæk er der tale om valg og derfor også fravalg, således at andre SAS-brugere ville kunne tænkes at løse tilsvarende problemer på anden vis.

Vejledningen baseres på SAS Version 6.12. Nyligt er der udgivet en Version 8.00, men langt overvejende er der ikke ændret på de ting, der behandles i vejledningen, hvorfor denne langt overvejende kan benyttes i forbindelse med arbejde i den nye version også. Således kan al vejledning i forbindelse med programmering benyttes umiddelbart i den nye version, og resultat-udskrifterne ser med ganske få undtagelser ligesådan ud. Nogle få steder i vejledningen er der dog indsat noter vedrørende ændringer.

Undertegnede er ansvarlig for vejledningens indhold, og for hvad der er udeladt, men for kommentering af tidligere udkast til vejledningen skal der lyde stor tak til Mette Tobiasen og Martin Munk, begge på Institut for Økonomi, Politik og Forvaltning, Aalborg Universitet, samt ikke mindst til Kamma Langberg, Analyseinstitut for Forskning.

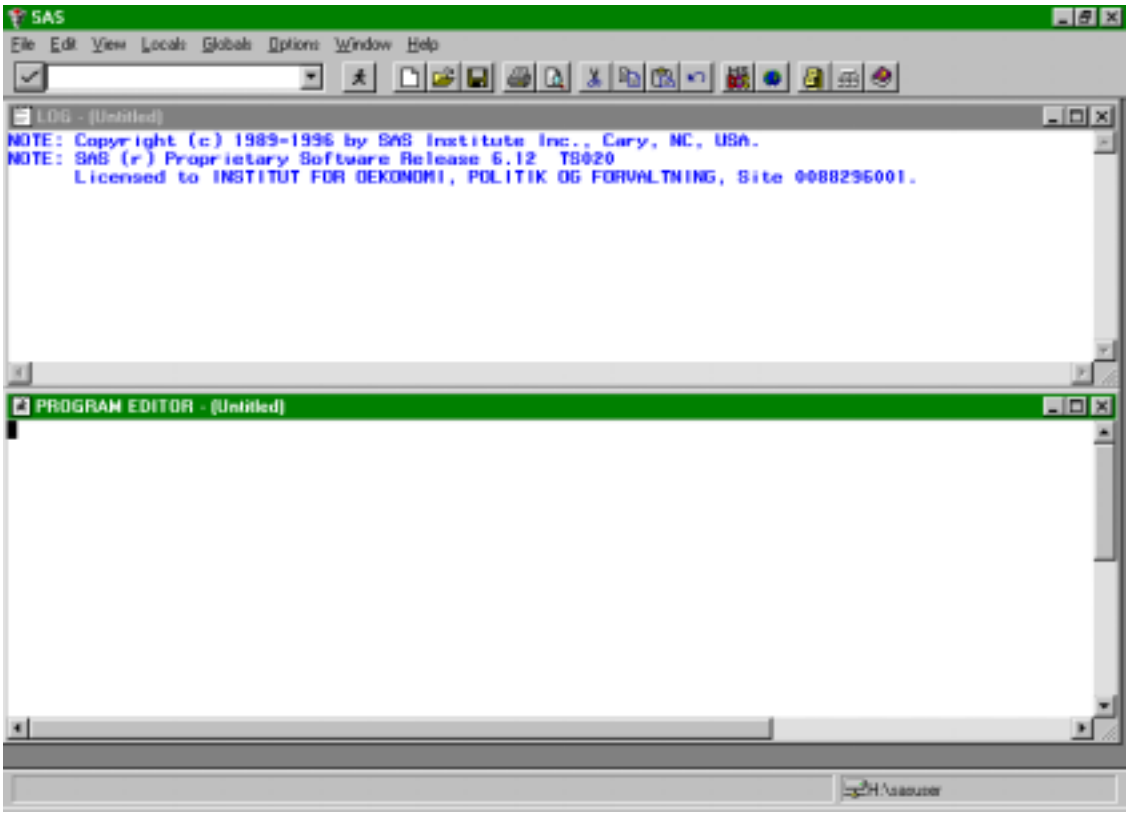
Henrik Lolle

1. AT FINDE RUNDT I SAS

Jeg vil i dette indledende kapitel se på den grundlæggende struktur i de dele af SAS, som bliver behandlet i vejledningen. I kapitel 2 og 3 beskrives derpå, hvad et datasæt og et SAS-program er for noget, og først derefter vil jeg gå over til den egentlige vejledning i data-tilgang, -behandling og -analyse.

Grundstrukturen er bygget op omkring tre vinduer. Disse vinduer samt brugerfladen ASSIST præsenteres i det følgende. Når man har klikket dig ind på SAS møder man straks to af disse vinduer, nemlig LOG-vinduet i øverste halvdel af skærmen og PROGRAM EDITOR-vinduet i den nederste, sådan som det er vist på Skærmprint 1 herunder (det *kan* sættes op anderledes, men det vil normalt være sådan, man ser det første gang).[2]

Skærmprint 1



[2] I SAS Version 8 findes endnu et vigtigt vindue, der under såkaldt 'default'-indstilling ligeledes viser sig i åbnings-billedet. I venstre side findes her en 'Explorer', hvor der kan ses, hvilke biblioteker, der er til rådighed, og hvilke resultater, der er udskrifter af, og det er f.eks. nemt herfra at 'springe' ind i specifikke datasæt eller resultatudskrifter.

Udover programeditoren og logskærmen findes der et OUTPUT-vindue, hvor alle ens analyse-resultater vises, og hvis man farer vild i andre dele af SAS, kan man altid finde tilbage til disse tre vinduer ved at trykke på følgende funktionstaster: F5 for PROGRAM EDITOR, F6 for LOG og F7 for OUTPUT. En oversigt over alle de enkelte funktionstaster kan i øvrigt findes ved at skrive '*keys*' i kommando-linien (den lille skriverubrik øverst til venstre i åbningsvinduet), hvorefter der trykkes på '*return-tasten*'.

De enkelte vinduer kan i øvrigt sættes til enten at fylde hele skærmen, en del heraf eller figurere som en lille bjælke nederst på skærmen - der klikkes blot på en af knapperne i vinduets øverste højre hjørne.

Øverst i skærmbilledet, over LOG-skærmen findes dels en række ikoner, dels en række menuer. Menuerne kan også til enhver tid fremkaldes ved højreklik på musen, i form af en såkaldt *pull down* menu. Jeg vil ikke her beskrive alle mulighederne slavisk, men blot referere til de muligheder, der bliver brug for undervejs i forbindelse med eksemplerne.

LOG-vinduet:

LOG-vinduet fungerer som loggen på et skib - alt, hvad der sker, beskrives her. Hvis der køres analyser med "peg og klik"-brugerfladen ASSIST, så angiver SAS de underliggende programstumper her, og hvis man kører egne programmer fra programeditoren, så gentages disse i loggen med eventuelle fejlmeddelelser (med rød skrift), advarsler (med grøn skrift) eller andre oplysninger (med blå skrift). Fejlmeddelelserne henviser ofte til *syntaksfejl*, dvs. til fejl i program*strukturen* eller i de enkelte "program*ord*", men kan også henvise til en ulovlig division med '0' f.eks. Advarsler med grøn skrift, som altså ikke nødvendigvis signallerer fejl, kan f.eks. dreje sig om, at programmet har resulteret i beregning på tomme felter, og dette resulterer i, at resultatvariablerne for de pågældende observationer også bliver tomme. Efter en advarsel herom er det så op til én selv at bedømme, hvorvidt programmet er ok alligevel.

PROGRAM EDITOR-vinduet:

PROGRAM EDITOR-vinduet er stedet, hvor man nedskriver, gemmer, retter og kører sine egne programmer. Det kan dreje sig om programmer, der sætter analyser i gang og printer resultaterne ud, eller det kan dreje sig om såkaldte rekodninger eller andre manipulationer af dataene for at tilpasse dem analysen. Rekodninger er man nødt til at lave i programeditoren, men de fleste analyser kan foretages fra brugerfladen ASSIST (se efterfølgende)[3]. Der er imidlertid store fordele forbundet ved at analysere fra program-

[3] Simple rekodninger kan klares gennem klik og peg i den nye SAS Version 8.

editoren. For det første kan man, så snart man har fået lavet én analyse i programform, lynhurtigt reproducere denne med simpel kopiering og indsættelse, som det kendes fra f.eks. tekstbehandlingsprogrammer. Herefter kan programmet justeres, hvorefter der foretages en anden, men lignende analyse - f.eks. blot med andre variabler. På den måde kan man altså hurtigt få opbygget et program til mange analyser. For det andet kommer man ofte ud for, at man på et senere tidspunkt skal foretage de samme analyser igen, enten helt nøjagtigt som de er kørt tidligere, eller med enkelte justeringer - f.eks. kun med en delmængde af stikprøven eller med udprintning af nogle ekstra statistiske test. Af disse årsager vil det derfor ofte være *meget* tidsbesparende at arbejde i programeditoren fremfor i ASSIST.

Da det ofte kan være svært at få taget hul på programmeringen, bl.a. fordi man ikke nøjagtigt husker en given analyses programstruktur, vil det dog tit kunne betale sig at køre en analyse i ASSIST til at starte med og så kopiere det bagvedliggende program fra LOG-vinduet og efterfølgende sætte det ind i programeditorer. Dette gøres ved markering, kopiering og indsættelse. Man skal så dog efter indsættelse af programmet i programeditoren sørge for at slette program*numrene* i starten af hver linje. Alternativt - og måske lidt nemmere - kan man efter kørsel i ASSIST gå direkte til programeditoren og derefter klikke på funktionstasten F4. På den måde indsættes programmet i program-editoren, men for overskuelighedens skyld kan det i så fald anbefales at slette den store mængde af overflødige linjer, som SAS printer med (program-eksemplerne i nærværende vejledning skulle gøre det nemt for læseren at bedømme, hvilke linjer der er væsentlige).

Programmerne gemmes som i andre windows-baserede software-pakker, gennem 'file' – 'save'/'save as' eller ved klik på diskette-ikonet. Det kan alt sammen lyde en smule indviklet at skulle lave sine egne programmer i en tid, hvor man efterhånden er blevet vænnet til peg og klik-brugerflader, men i praksis er det faktisk meget simpelt. Og der er, som nævnt, store fordele knyttet til analyse fra programeditoren.

Der skal her lige nævnes to væsentlige detaljer angående programeditoren. Hvis man i tekstbehandlingssystemer som WORD eller WP markerer noget af en tekst, så vil markeringen forsvinde, hvis man bevæger cursoren med piletasterne. Dette sker *ikke* i SAS[4]. Markeringen bliver stående, og hvis man nu begynder at skrive ny tekst, så forsvinder det markerede (men kan selvfølgelig genkaldes ved klik på fortryd-ikonet). Sørg derfor for at klikke en gang med musen, inden du begynder at skrive igen efter markering - med mindre selvfølgelig at det er meningen, at den markerede tekst skal erstattes. En anden forskel mellem tekstbehandlingsprogrammer og programeditoren i SAS er den måde, hvorpå filer åbnes. I tekstbehandlingssystemer er vi vant til at kunne

åbne flere filer/dokumenter, som så præsenteres i forskellige vinduer, man kan skifte mellem. I SAS er der kun ét programeditor-vindue til rådighed ad gangen, og derfor åbnes en ny fil (et SAS-program) oven i det, der allerede ligger der, sådan at de to programmer kommer til at ligge *i forlængelse* af hinanden i samme vindue med det senest åbnede først og med dettes navn[5]. Så hvis man vil kalde et andet program frem end det, der ligger på skærmen, uden at disse to programmer bliver rodet sammen, skal den programtekst, der ligger der i forvejen, altså først slettes. Dette gøres ved at klikke 'Edit' - 'Clear text' enten fra menulinjen for oven eller via *pull down*-menuen (husk at gemme programmet først, hvis der er ændret i det).

OUTPUT-vinduet:

OUTPUT-vinduet er som nævnt stedet, hvor resultaterne vises. Dette "popper" op, når man via enten programeditoren eller ASSIST foretager statistiske analyser. Man kan så studere resultaterne og eventuelt kopiere dem og sætte dem ind i et tekstbehandlings-program som f.eks. WORD eller WP. Kopieringen foretages ved at klikke på *edit - select all* eller bare blokke en del af resultaterne. Disse indsættes nu blot i et dokument i tekstbehandlingssystemet. Hvis ikke den indsatte tekst automatisk fremstår, som den så ud i OUTPUT-vinduet i SAS, så markér det indsatte, gå ind i *skrifttype* og vælg *SAS Monospace* punktstørrelse 10. Husk blot at SAS-tabeller som oftest ikke vil kunne bruges direkte til præsentation i rapporter. Enten er man nødt til at ændre dem en del, eller også benytter man slet ikke SAS-udskrifterne i præsentationen. I stedet for opbygges helt nye tabeller i tekstbehandlingssystemet eller i et regnearksprogram (som f.eks. EXCEL), hvor kun de relevante tal præsenteres. Sidstnævnte alternativer er de hyppigst anvendte. Men ved store krydstabeller f.eks. kan det af og til betale sig at overføre SAS-tabellerne til eksempelvis EXCEL. Hvordan man rent praktisk gør dette, vises senere i et eksempel.

Outputresultaterne kan også gemmes inde i SAS gennem 'File' - 'Save as'. Sært nok kan man tilsyneladende ikke åbne filen med gemte resultater fra output-vinduet, men i SAS kun fra programeditoren - gennem 'File' - 'Open'. Vær i øvrigt opmærksom på, at analyseresultaterne ikke altid kan være på ét skærmbillede, og at man "ankommer" til output-vinduet nederst i rækken af resultater. Når output-vinduet "popper" op, er det derfor en fordel med det sammen at gå til toppen ved at *schrolle* med bjælken til højre eller ved at klikke på 'Home'-knappen, mens 'Ctrl'-knappen holdes nede.

---

[4] Dette er imidlertid ændret i SAS Version 8.

[5] I Version 8 kan der åbnes flere forskellige programmer på samme tid.

"Peg og klik"-brugerfladen ASSIST:

Oprindeligt var der kun programeditoren at arbejde fra i SAS, men det er nu muligt via ASSIST med ganske få museklik at foretage en lang række statistiske analyser. Man kommer ind i brugerfladen ved at klikke på ASSIST-ikonet øverst på skærmen - den viser tre rækker knapper med en stor firkant ovenover (nr. to knap på Skærmprint 1) [6]. Man kommer nu til *Primary Menu*, hvorfra der kan vælges mellem en række undermenuer. Klikkes der på en undermenu-knap, kommer man et skridt videre ind i træstrukturen, ved at undermenuen udspecificeres gennem visning af en række nye knapper. På den måde fortsættes, indtil man når en "grenspids", hvor de endelige specifikationer angående analysen foretages - hvilket datasæt, hvilke variabler, hvilket output er der behov for osv. Man kommer tilbage gennem træstrukturen ved at klikke på en *Goback*-knap eller på krydset i vinduets øverste højre hjørne.

Ved de fleste af vejledningens eksempler, med SAS-programmer til rekodning eller analyse, angives kort, hvordan noget tilsvarende kan laves i ASSIST. Det er imidlertid langt fra alt, der kan laves her, så for de fleste brugere vil det være nødvendigt at lære lidt programmering også - og som nævnt kan der også tit spares en masse tid ved blot at starte i ASSIST for så efterfølgende at fortsætte i programeditoren. Og én meget stor fordel ved at arbejde i programeditoren er, at man kan gemme sine kørsler. En vigtig forskel mellem kørsler i ASSIST og programeditoren er i øvrigt, at mens output fra ASSIST automatisk skifter tidligere output ud, så bliver output fra kørsler i programeditoren blot lagt i enden af det tidligere output. Hvis man ikke er interesseret i, at al outputet hober sig op, så må man selv sørge for at slette det, inden man går tilbage til programeditoren og kører nye statistiske analyser. Dette gøres ved at klikke 'Edit' - 'Clear text' (enten fra menuen øverst i skærmbilledet eller gennem *pull down*-menuen, som fremkommer ved et højreklik på musen) eller ved at taste 'Ctrl' og 'c' på samme tid. Sådan slettes i øvrigt tekst i programeditoren og logskærmen også.

I den ASSIST-vejledning, der følger umiddelbart efter en del af afsnittene, vil det visse steder forudsættes, at man selv tænker sig til nogle mellemliggende trin. Dette gælder f.eks. alle de steder, hvor der skal vælges datasæt til analysen, samt de variabler, der specifikt skal analyseres på. Skrives der f.eks. "Vælg 'Active data set' og 'Variables' ", så betyder det, at der skal museklikkes på knappen 'Active data set', hvorpå der fremkommer et vindue med de datasæt, der er til rådighed. Der skal så klikkes på det datasæt, man er interesseret i. Og derpå skal der klikkes på knappen 'Variables', så på

[6] Der findes andre brugerflader i SAS. I Kapitel 9 beskrives ganske kort INSIGHT. Det nyeste skud på stammen af brugerflader er ANALYST, som vanligvis ligger fast i den nye Version 8. I Version 8 ligger indgangen til ASSIST i øvrigt ikke længere som en ikon. Her skal man klikke på den nye menu 'Solutions' og derpå 'ASSIST.

den eller de relevante variabler og til slut på knappen ‘OK’. Det fungerer på stort set samme facon i alle procedurer, så det ville være meningsløst (og måske endda meningsforstyrrende) at skrive det helt ud i alle eksemplerne.

Når man vil tilbage til det oprindelige billede med LOG og PROGRAM-EDITOR, kan der enten trykkes på funktionstasterne F6 og F5 eller blot klikkes på krydset i øverste højre hjørne på ASSIST-vinduet. Vær opmærksom på, at det øverste kryds tilhører selve SAS, så hvis der klikkes her, spørger SAS, om man virkelig mener, at programmet skal lukkes ned.

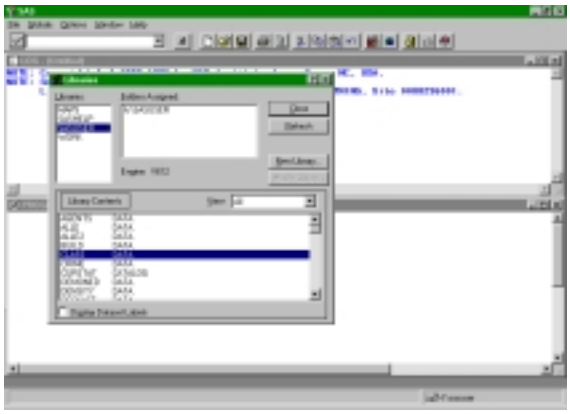
2.SAS DATASÆT

Det forudsættes egentlig, at læsere af denne vejledning har en grundlæggende viden om, hvad et datasæt er for noget, samt hvilke elementer et sådant består af, f.eks. gennem læsning af den tidligere nævnte bog af Peter Nielsen. I denne bog beskrives det bl.a. også, hvordan man gennem ASSIST etablerer forbindelse med datasæt. Jeg vil dog ganske kort rekapitulere disse ting i det følgende.

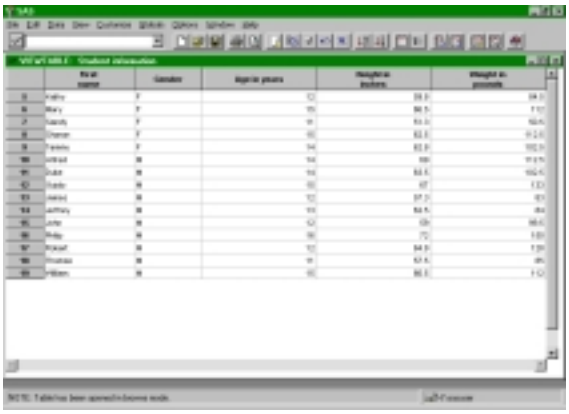
2.1. Hvad er et SAS-datasæt?

Et datasæt i SAS er udformet som en todimensionel matrice (tabel)[7]. Prøv selv at få et overblik over et datasæt ved at klikke på kartoteks-ikonet i SAS (tredje knap fra højre på Skærmprint 1) - man kan komme ind i en lidt anderledes opsætning af et datasæt gennem ASSIST også[8], hvor man kan få vist enten hele matricen eller en enkelt observation. Når man har klikket på kartoteks-ikonet, "popper" der et lille vindue op, hvori vises de forskellige SAS-biblioteker og derunder liggende datasæt, der umiddelbart er til rådighed (se skærmprint 2). Ofte vil der automatisk vises de datasæt, som ligger under biblioteket *sasuser*. Om og hvad der ligger af data, afhænger af hvordan SAS er installeret, og hvilken version af SAS der er tale om, men der burde ligge en række øvesæt til rådighed. Dobbelt-klik på et tilfældigt datasæt, og der vises en Excel-lignende matrice, datasættet (se skærmprint 3).

Skærmprint 2



Skærmprint 3



[7] Der findes flerdimensionelle tabeller i SAS MDDB, men de vil ikke blive berørt her.
[8] Se f.eks. hvordan i Nielsen (1999), s. 173-74.

Hver enkelt række er én *observation*, og hvis det er almindelige survey-data, vil denne indeholde besvarelserne (og evt. andre informationer) fra en enkelt respondent. Hvis der f.eks. er 1.500 rækker nedad, har vi altså fat i en survey-undersøgelse med besvarelser fra 1.500 respondenter. Ud ad mod højre ligger rækken af *variabler*, og hvis vi igen taler om survey-data, så vil en variabel typisk svare til ét svar[9]. Datasættet kan sandsynligvis ikke være på én side, som det viste eksempel, og man 'schroller' så enten nedad eller udad mod højre for at se resten.

Datasættet vises med *labels* - en variabelbeskrivelse - og ikke variabel*navne*, som højst må fylde otte karakterer (og som skal begynde med et bogstav)[10]. Hvis man hellere vil have vist variabelnavnene, så klikkes blot på: 'View' - 'Collumn Names'. Og vil man have mere detaljerede oplysninger om en enkelt variabel, så dobbeltklikkes på det grå felt med label eller navn, der hører til pågældende variabel. Her kan f.eks. ses, hvilket format der er knyttet til variablen. Formatet bestemmer, hvordan variablen skrives i output; det kan eksempelvis være et rent numerisk format med to decimaler, som ligger fast i SAS, eller et selvfremstillet format, der skriver "ja" for "0" og "nej" for "1"[11]. Hvis man gør datasættet klar til redigering ved at klikke 'Edit' - 'Edit Mode', er det også muligt at slette enten variabler eller observationer, rette i enkelte observationer, sortere datasættet osv., og man kan gemme det manipulerede datasæt under *File*, evt. med et nyt navn (sørg i hvert fald altid for at have *back-up* af det oprindelige datasæt; det vender jeg tilbage til senere). Vejledning til disse ting er ikke det primære formål her, og det er desuden nogenlunde simpelt at finde ud af, så jeg går skyndsomt videre med at beskrive forskellige variabeltyper og derpå til det centrale i vejledningen: forberedelse og analyse.

Typer af variabler i et datasæt:

I forbindelse med variabler og de forskellige værdier, disse variabler kan antage, taler man om forskellige måleniveauer. Forskellige typer af variabler har forskellige måleniveauer, og dette skel er meget vigtigt i forbindelse med data-analysen. Til mange analyse-metoder er der nemlig krav om bestemte måleniveauer.

---

[9] Dog vil der ofte være variabler med andre oplysninger også. Således vil der næsten altid være en variabel med et id-nummer, som modsvarer det nummer, der står på pågældende spørgeskema. Hvis der er tale om en landsdækkende undersøgelse, vil der ofte også være en variabel indeholdende kommunenummer til identifikation af den kommune, respondenten bor i. Og der vil kunne være tale om en række andre oplysninger, som ikke stammer fra respondent-svar, men som undersøgeren har tilføjet - uden konsekvenser for anonymiteten selvfølgelig. Herudover vil der efterhånden, som der arbejdes på datasættet, oprettes en række nye variabler, dannet ud fra de oprindelige.

[10] Reglerne for navngivning er mindre strenge i Version 8.

[11] Se f.eks. herom i Nielsen (1999), s. 158-62.

Det simpleste måleniveau er nominalskala. Her kan man blot klassificere de forskellige værdier, men man kan ikke rangordne dem. Et banalt eksempel kan her være et spørgsmål om, hvilken farve man bedst kan lide: rød, grøn eller blå. Svaret kan enten være rød, grøn eller blå (eller evt. ved ikke), og svarene kan derfor klassificeres, men der findes ikke én naturlig måde at rangordne disse svar på - rød er f.eks. ikke naturligt hverken større eller mindre end blå, og heller ikke naturligt bedre eller værre end blå osv. Et andet eksempel er en variabel for køn. Her er kategorierne mand og kvinde, og heller ikke disse har en bestemt rangorden[12].

Det næste måleniveau er ordinalskala. Her kan kategorierne rangordnes, og der kunne f.eks. være tale om en variabel for et surveyspørgsmål om, hver enig eller uenig respondenten er i et udsagn. Der kan eksempelvis være afsat fem faste svar-kategorier: 'Meget enig', 'noget enig', 'hverken enig eller uenig', 'noget uenig' og 'meget uenig'. Det kan selvfølgelig diskuteres, hvordan svar-rækken skal vende, dvs. om de skal vende som her eller omvendt, men bortset herfra kan de ikke naturligt skrives op på anden vis.

Derefter kommer intervalskala, hvor værdierne ikke blot er rangordnede, men hvor der også naturligt hører bestemte værdier til kategorierne, således at afstanden/intervallet mellem to 'nabo'-kategorier altid er den samme. Der er dog intet naturligt nulpunkt i skalaen, og man kan derfor ikke beregne forhold mellem værdierne. Det er ikke nemt at finde samfundsrelevante eksempler, og det hyppigst nævnte eksempel er sikkert en variabel for 'varmegrader'. Her kan der som måle-enhed f.eks. benyttes både Fahrenheit og Celcius, og disse skalaer har forskellige nulpunkter. Man kan f.eks. ikke sige, at 20 grader Celcius er dobbelt så varmt som 10 grader Celcius. Vær opmærksom på, at intervalskala *ikke* betyder, at de enkelte kategorier er inddelt i intervaller - en nærliggende misforståelse - men altså derimod at der er lige store intervaller *mellem* kategorierne.

Til slut findes ratioskala eller forholdstalsniveau, hvor der foruden egenskaberne ved intervalskalaen også findes et naturligt nulpunkt. Ofte vil man imidlertid ikke skelne mellem ratio- og intervalskalerede variabler, da stort set de samme statistiske analyser kan udføres på begge typer - eksempelvis lineær regression. Et eksempel på en ratioskaleret variabel kunne være personlig indkomst, hvor f.eks. 400.000 kr. er dobbelt så stor en indkomst som 200.000 kr. Vær dog opmærksom på, at netop en variabel for

[12] Selvom en variabel for køn defineres som nominalskaleret, kan den ligesom alle andre dikotome, nominelle variabler i analyser behandles, som om der var tale om en interval- eller ratioskaleret variabel. Dette hænger sammen med, at det væsentlige karakteristikum ved interval- og ratioskalerede variabler i forbindelse med analyser er, at afstanden mellem kategorierne ikke er forskellig (se herom nedenfor), og da der i dikotome variabler kun findes én afstand, kan dette jo siges at være tilfældet.

personlig indkomst ofte vil være målt på ordinalskala og ikke hverken interval- eller ratioskala. For det første vil man blot bede respondenten om at sætte et kryds inden for en række på forhånd opstillede indkomst-intervaller. For det andet vil disse intervaller typisk ikke være lige store, sådan at der heller ikke er lige stor afstand mellem de enkelte svar-kategorier (eller disses centrum). I forhold til spørgsmålet om hvilke statistiske analyser der kan foretages med sådan en variabel, er det første problem det mindste, og i praksis er alle variabler jo alligevel indelt i intervaller, selvom de kan være nok så små. Selv i forbindelse med en variabel som alder vil man ikke bede respondenten om at angive denne helt præcist - f.eks. 'treogtredieve syv ottendedele år', men blot bede om et heltal, altså 'treogtredieve år'. Blot der findes tilstrækkeligt mange intervaller, kan det uden problemer forsvares at benytte sådanne variabler i forbindelse med en række statistiske analysemetoder, der formelt kræver interval-/ratio-skalerede variabler. Det andet forhold - at der ikke er lige stor afstand mellem de enkelte kategorier - er et større problem, hvilket dog langt fra afholder samfundsforskere fra at foretage statistiske analyser, hvortil der mindst kræves intervalskala. Dette skal imidlertid ikke diskuteres nøjere her. Blot skal det pointeres, at det som et minimum forventes, at man er opmærksom på de problemer, der kan være forbundet med formelt ukorrekt brug af analysemetoder.

Variabler fra de to førstnævnte skalaniveauer - nominal- og ordinalskala - kaldes også for diskrete variabler, fordi de kun er inddelt i et afgrænset antal kategorier/værdier, mens variabler fra de sidste to - interval- og ratioskala - benævnes kontinuerte variabler, da der ikke er nogen grænse for, hvor fint værdierne kan inddeles. Jeg senere i forbindelse med de enkelte statistiske metoder komme ind på, hvilke måleniveauer der formelt kræves til disse.

2.2. Tilgang til datasæt

Fra SAS-systemet er der ikke umiddelbart adgang til alle de SAS-datasæt, som ligger på ens computer eller det net, man er tilsluttet. Kun de datasæt, som ligger på biblioteker/mapper[13], der automatisk skabes reference til ved åbning af SAS-programmet - f.eks. SASUSER - kan bruges umiddelbart. Hvis man derfor har indsat eller oprettet sit eget datasæt i et andet bibliotek, hvortil der ikke skabes automatisk reference, er man nødt til at udpege dette bibliotek for SAS[14]. Og hvis der er tilhørende formater, skal der

[13] Bibliotek og mappe er synonymer. SAS benytter termen 'bibliotek',
[14] Se også herom i Nielsen (1999), s. 156-58.

ligeledes udpeges et formatbibliotek (som dog gerne må være det samme). Dette gøres meget simpelt på følgende facon i programeditoren[15]:

```
* Programeksempel 2.1;
libname SKOLE 'H:\DATA';
libname library 'H:\DATA';
```

Første sætning er en såkaldt kommentarlinie. Kommentarlinier kan indsættes, hvor det skal være i et program, og SAS tager ikke notits af dem. De er der udelukkende for overskuelighedens skyld. Kommentarlinier skal begynde med en stjerne (til højre for 'ø' på tastaturet) og slutte med et semikolon. Her kan man skrive, hvad programmet laver, sådan at man senere kan finde rundt i det. Dette er især vigtigt, når der efterhånden bliver skrevet mange programlinier eller måske -sider.

Anden sætning i eksemplet udpeger et datasæt-bibliotek, og tredie sætning udpeger et format-bibliotek. Der skal selvfølgelig ikke nødvendigvis skrives nøjagtigt det samme som i eksemplet. Det kommer helt an på, hvad man har kaldt det bibliotek, hvorunder ens data og formater ligger. I dette tilfælde ligger begge dele på 'H:\DATA'. Ordet 'SKOLE' i anden sætning er et selvvalgt ord, som skaber en reference til biblioteket 'DATA' med tilhørende sti (det kaldes i SAS-sprog for *libref* - *lib*rary *ref*erence). Dvs. hver gang man efterfølgende, henviser til 'SKOLE', så ved SAS, at der er tale om stien 'H:\DATA' - det er altså en slags kaldenavn. Men husk at referencen til biblioteket skal genoprettes, hver gang SAS-systemet åbnes (og man kan så i øvrigt kalde det noget nyt, hvis man skulle have lyst til det - det er jo blot et kaldenavn eller alter ego, som henviser til den nøjagtige fysiske adresse). At man på den måde skal fortælle SAS, hvilke biblioteker, men vil kunne hente datasæt og formater fra, kan forekomme lidt underligt, for i de fleste andre softwarepakker som f.eks. WORD og WP skal man ikke udpege sådanne[16]. Men dette *skal* altså gøres *hver gang man kommer ind i SAS*, med mindre man har både datasæt og formater liggende under biblioteket SASUSER. Ordet 'library' i tredie sætning er reserveret af SAS, og det henviser altid til det bibliotek, hvor man har placeret sine formater. 'Library' er altså en fast reference til den fysiske adresse, hvor ens formater ligger. Vær i øvrigt opmærksom på, at når der laves formattilknytninger, så skal der skabes reference til formatbiblioteket uanset om dette er SASUSER-biblioteket. Der bliver nemlig kun skabt automatisk reference til SASUSER-biblioteket som et data-

[15] I alle vejledningens programeksempler vil selvvalgte navne for biblioteker og variabler være skrevet med kapitæler. SAS er i virkeligheden ligeglad med, om man benytter store eller små bogstaver, så dette er alene gjort for, at læseren kan skelne især variabelnavne fra de faste 'SAS-ord'.

bibliotek - ikke som et format-bibliotek. Man kan imidlertid selv sørge for, at automatisere dette - se herom nedenfor.

Når de tre program-linier er skrevet (eller kun nummer to eller nummer to og tre, alt afhængigt af om man vil have kommentarer med, samt om man har knyttet selvfremstillede formater til variablerne, så markeres disse med mus eller piletaster, og der klikkes på 'run'-ikonen. På Skærmprint 1 er det den første knap fra venstre, som viser en løbende person. Man *behøver* ikke at markere linierne, men der er flere fordele ved at gøre det inden kørsel af et program: For det første har man ofte flere programstumper og ønsker ikke at køre det hele. Ved markering køres kun det markerede. For det andet forsvinder alle programlinier fra programeditoren, når man kører uden markering. Ganske vist kan man ved tryk på F4-funktionstasten genkalde linierne, men hvis man i mellemtiden f.eks. har været inde i ASSIST og lavet en analyse herfra, så vil det være de bagvedliggende programlinier til denne analyse, der fremkaldes. Så et par gode råd: Gem programmer ofte; saml programdele i større 'klumper', så de danner en logisk struktur, der er nem at finde rundt i; og markér den del af programmet, der skal udføres, inden der klikkes på 'run'-ikonen.

Udpegning af bibliotek(er) behøver ikke nødvendigvis at ske via program-editoren. Man kan alternativt gøre det via det lille vindue 'Libraries', som vælges ved at klikke på kartoteksikonen (se Skærmprint 2) eller via ASSIST. Hvis man vælger at gøre det i 'Libraries' (som er det letteste), klikkes der på knappen 'New library'. Derpå vises et nyt lille vindue med plads til dels at skrive sit kaldenavn i boksen 'Library', dels at skrive den fulde sti i boksen 'Folder to assign'. Dernæst klikkes på 'Assign', og husk at gentage processen, hvis der skal skabes reference til et formatbibliotek (i så fald skrives ordet 'library' i boksen 'Library'). I dette 'Assign'-vindue kan man i øvrigt fortælle SAS, at man fremover vil have skabt den pågældende reference automatisk, hver gang man åbner SAS. Dette gøres ganske simpelt ved at klikke i boksen med teksten: 'Assign automatically at startup'. Der kan imidlertid kun skabes reference til ét formatbibliotek ad gangen, så hvis man automatisk skaber reference til et formatbibliotek og efterfølgende skaber reference til et andet, så forsvinder referencen til det første, og den automatiske reference er herefter sat ud af kraft.

[16] Det hænger sammen med, at SAS-programmer uanset platform (dvs. styresystem og maskine) er ens, og da man også skal kunne afvikle SAS-programmer ved kørsler i baggrund på stort anlæg, hvor man skal lave sådanne henvisninger, kommer man altså også til det, selvom man sidder med sin egen hjemme-PC.

***Hvordan man gør i ASSIST (data-tilgang)*:**
Fra 'Primary Menu' vælges menuen 'SETUP'; derpå 'SAS data libries' og til slut 'Assign a new libref'. I den øverste linie skrives det navn, som man vil bruge som biblioteksreference, og i den nederste (tryk på tabulatortasten for at komme derned) skrives den fulde sti til den fysiske adresse, hvorpå der klikkes 'ok'. Så klikkes der 'Go back' to gange og derpå 'Assign format libref' (hvis der er selvfremstillede formater knyttet til en eller flere variable). Her skal der kun skrives stien, fordi ordet 'library' som nævnt er en af SAS fastlagt reference til formatbiblioteker. Til slut klikkes på 'ok' og igen på 'Go back' to gange, sådan at man vender tilbage til 'Primary Menu'.

3.HVAD ER ET SAS-PROGRAM?

Der vil i eksemplerne i vejledningen blive anvendt tre forskellige former for programmer, og disse vil kort blive beskrevet herunder. Der må dog gøres opmærksom på, at de kun er beskrevet og vist i en simpel form. For mere detaljeret og nøjagtig beskrivelse henvises til SAS-manualer.

Konkrete eksempler på programmer vil blive givet hele vejen gennem vejledningen. Yderst vigtigt i forbindelse med sådanne programmer er, at de er skrevet nøjagtigt i overensstemmelse med de syntaksregler, der gælder i SAS. F.eks. at alle sætninger afsluttes med semikolon, at man husker punktummet mellem biblioteks-reference og datasæt-navn osv. Der er dog ret frie hænder med hensyn til *layout*-strukturen, så man kan f.eks. fint skifte linie halvt inde i en SAS-sætning og indsætte diverse tabulator-indryk, hvor man vil. Det er også underordnet, om man benytter små eller store bogstaver eller begge dele. Men forsøg at få indarbejdet en standard for, hvordan programmerne opsættes, sådan at de nemt kan overskues.

- "*Styrekort*" er enkeltstående programsætninger, hvori der gives SAS forskellige oplysninger og anvisninger. Vigtigst er at fortælle, hvor de datasæt, som man vil arbejde med, befinder sig (et eksempel herpå er allerede vist i kapitel 2). Derudover kan disse sætninger eksempelvis dreje sig om, hvilket sideformat der skal benyttes i output, eller hvilket sidenummer der skal startes med. Følgende programsætning kan f.eks. bruges, hvis der er 'gået kuk' i ens sideformat:

  ```
  options linesize=94 pagesize=58 date number pageno=1;
  ```

  Sætningen sørger for, at de følgende udskrifter bliver med 94 karakterer pr. linje og 58 linjer pr. side, hvilket skulle være *default*. Endvidere startes forfra med side nr. 1 i output'et. Hvis man f.eks. er interesseret i et andet side-format, kan man selvfølgelig ændre tallene. Alternativt til programsætningen kan man klikke 'File' - 'Print setup' og ændre formatet herfra. Som oftest vil man blot benytte denne type af programlinier i starten af en SAS-session til at skabe reference til ens biblioteker.

- Et '*data step*' er et program, hvor man rekoder variabler, danner indekser eller foretager sig andre manipulationer af dataene. Først specificeres med en 'data-sætning', hvilket navn man vil kalde det færdig-manipulerede datasæt, samt med en 'set-sætning' hvilket datasæt man vil benytte som grundlag for manipulationerne - altså et output- og et input-datasæt. Dernæst følger selve de sætninger, hvor de forskellige beregninger og manipulationer foretages. Disse sætninger gennemløbes én gang for hver enkelt observation. Det vil altså sige, at hvis man eksempelvis danner en ny variabel, der er lig med summen af en række allerede eksisterende variabler, så foretager SAS disse operationer for hver eneste observation i datasættet. Sidst i programmet skrives en 'run-sætning'.

- En '*procedure*' er et program, der har en mere regelfast struktur end et datastep. Ofte benyttes procedurer til at udføre statistiske beregninger og (almindeligvis) udprinte resultaterne herfra, f.eks. almindelige frekvenstabeller eller korrelationsmatricer (resultaterne kan evt. lagres i et nyt datasæt). Men der findes også andre former for procedurer - f.eks. procedurer, der foretager systematiserede manipulationer på datasættet som f.eks. sortering, og procedurer, der danner såkaldte 'formater', der benyttes i forbindelse med formatering af output. Der findes en lang række procedurer indbygget i SAS med en fast struktur, og som alle i deres program-kode begynder med ordet 'proc' efterfulgt af procedurens navn, f.eks. 'freq' eller 'mean', og ved hver enkelt procedure findes en række forskellige muligheder for at specificere helt præcist, hvad det er, der skal foretages.

4. UNIVARIAT ANALYSE

Vi skal nu i gang med selve analysen af data. Kapitel 4 handler om den indledende deskriptive del af analysen - den del, hvor man ser på de enkelte variabler én ad gangen. Derefter følger et kapitel, hvor det gennemgås, dels hvordan man manipulerer data, så de egner sig bedre til den efterfølgende analyse af sammenhænge mellem variabler, dels hvordan man kan forbedre output, så det bliver mere læsevenligt. I kapitel 6 og 7 ser vi på forskellige måder, hvorpå man kan analysere sammenhænge mellem to eller flere variabler (koncentrationen vil her ligge på krydstabeller og korrelationskoefficienter), og endelig i kapitel 8 vises, hvordan man kan danne såkaldte indeks og typologier - komprimerede udtryk for to eller flere variabler.

4.1. Beregning af forskellige statistiske mål - minimum, maksimum, gennemsnit, standardafvigelse osv.

Én måde at analysere enkeltvariabler på, er at beregne statistiske mål, som hver især rummer en del information om variablen, men som klart nok også udelukker en masse information. Det er mål som f.eks. gennemsnit og standardvariation. I almindelige surveyundersøgelser vil langt de fleste variabler være på ordinalskala-niveau, og det kan derfor være problematisk at beregne f.eks. gennemsnit, varians og standardvariation, der forudsætter mindst intervalskalaniveau, mens mål som minimum, maksimum og variationsbredde er uproblematiske. Ofte vil man dog vælge at få beregnet statistiske mål, der formelt set ikke er tilladte - f.eks. gennemsnit på en ordinalskaleret variabel. Sådanne "ulovligheder" kan give nyttig information på trods af fejlbehæftelsen, og hvis der f.eks. er dannet sumindeks, hvor flere variabler er summeret (se herom i Kapitel 8), er det udbredt blandt samfundsforskere. Som det før er nævnt, er det imidlertid meget vigtigt at være (og gøre) opmærksom på disse kilder til fejl, når man så at sige skærer en hæl og klipper en tå for at få nogle vigtige informationer eller analyseresultater, man strengt taget ikke kan gøre fordring på.

En anden ting, man skal være opmærksom på ved disse statistiske mål, er, at der som nævnt er udeladt en masse information i de enkelte tal. F.eks. er det sjældent nok at få beregnet gennemsnit, fordi dette mål ikke siger noget om spredningen og eventuelle skævheder i fordelingen. For at undgå at få en fejlagtig opfattelse af fordelingen af den pågældende variabels værdifordeling, er det nødvendigt at få beregnet flere *forskellige*

mål, som tilsammen kan give et tilstrækkeligt billede af fordelingen. Men hvor mange og hvilke mål, der skal beregnes afhænger selvfølgelig både af variablenes måleniveau og det konkrete behov.[17]

Der findes flere forskellige måder, hvorpå man i SAS kan få beregnet disse univariate statistiske mål. Der skal her omtales procedurerne *mean* og *univariate*. Vil man f.eks. have beregnet antal valide, antal *missing values*, minimum- og maksimumværdi, gennemsnit og standardafvigelse for de to variabler 'HEIGHT' og 'WEIGHT' fra datasættet 'ELEVER' med biblioteksreferencen 'SKOLE', kan følgende programlinier skrives:

```
*Programeksempel 4.1;
proc means data=SKOLE.ELEVER
  n nmiss min max mean std;
  var HEIGHT WEIGHT;
run;
```

I anden sætning - efter kommentar-sætningen - specificeres følgende: 1) Hvilken procedure der er tale om, her 'means'; 2) hvilket datasæt der skal analyseres på, her 'SKOLE.ELEVER'; samt 3) hvilke statistiske mål, der skal beregnes. I alle vejledningens eksempler specificeres det relevante data-sæt, men ofte vil dette være overflødigt. Hvis der ingen datasæt specificeres, vil det sidst aktiverede datasæt nemlig benyttes til analysen. Aktivering af et data-sæt kan ske via en procedure eller via et data-step. Læg i øvrigt mærke til, at nummer to sætning er delt på to linier, og at der er benyttet indrykning. Som nævnt tidligere, er sådanne layoutmæssige detaljer ikke uoverensstemmende med syntaksreglerne, og de hjælper én til at kunne overskue programmet.

Ud over de statistiske mål, der er bedt om i programeksemplet, kan der bedes om en serie andre. De fleste muligheder nævnes herunder:

[17] Se f.eks. Hellevik (1994), p. 195-210, for uddybning.

N (antal valide observationer)
NMISS (antal observationer med manglende værdi i pågældende variabel)
MIN (minimumsværdi)
MAX (maksimumsværdi)
RANGE (variationsbredde)
SUM (sum)
MEAN (gennemsnit)
VAR (varians)
STDERR (standardfejl for gennemsnit)
CV (variationskoefficient)
SKEWNESS (mål for skævhed i fordeling)
KURTOSIS (mål for hvor spids eller flad fordelingen er)
T (Student's t for test populationsgennemsnit på 0)
PRT (Sandsynlighed for ovennævnte test - altså populationsgennemsnit på 0)

SKEWNESS og KURTOSIS er relevante, hvis man vil checke, om variablens værdier er normalfordelte, men de kan være svære at fortolke. Som Sven Kreiner skriver: "De to udtryk for skævheden og kurtosen er udpræget utilnærmelige og de to størrelser anvendes da kun også sjældent i praksis" (Kreiner 1999, p. 52). Både graden af utilnærmelighed og de to størrelsers sjældne brug i praksis er dog ganske givet overdrevet her.

Hvis man har behov for at checke, hvorvidt værdierne stammer fra en normalfordelt population, kan i stedet benyttes *proc univariate*, som kan udregne sandsynligheden herfor. Dette gives der et eksempel på nedenfor. Bedes der ikke eksplicit om nogen statistiske mål, giver SAS jer default-målene, som er antal valide, minimum, maksimum, gennemsnit og standardvariation.

Hvis man har behov for at se forskellige statistiske mål på *sub-grupper* i stikprøven, kan der indsætte en *class*-sætning. Nedenstående programeksempel viser f.eks., hvordan de statistiske mål fra før kan beregnes for piger og drenge hver for sig (variablen for køn har navnet *SEX*, og sættes der flere *class*-variabler på, opdeles output i undergrupper af undergrupper):

```
*Programeksempel 4.2;
proc means data=SKOLE.ELEVER
  min max mean std;
  var HEIGHT WEIGHT;
  class SEX;
run;
```

Herved fremkommer følgende output:

SEX	N Obs	Variabler	Minimum	Maximum	Mean	Std Dev
D	187	HEIGHT	132.0000000	188.0000000	162.3208556	14.9199925
		WEIGHT	31.0000000	88.0000000	58.9465241	12.3064511
P	180	HEIGHT	132.0000000	191.0000000	160.5166667	14.5446625
		WEIGHT	33.0000000	90.0000000	57.1117318	11.4851554

Drengenes højde og vægt er gennemsnitligt en anelse større end pigernes, ligesom disse størrelser også svinger mere hos drengene (se Std Dev - standardvariation). Det fremgår imidlertid ikke af resultaterne, om disse forskelle er signifikante - dvs. om man f.eks. vil kunne konkludere, at drengene *generelt i populationen* er højere end pigerne. Hertil skal benyttes en form for variansanalyse, nærmere bestemt en simpel t-test (er ikke beskrevet i nærvarende vejledning)[18].

Hvordan man gør i ASSIST (univariate statistiske mål):
Vælg 'Primary menu' - 'DATA ANALYSIS' - 'ELEMENTARY' - 'Summary statistics'. Derpå vælges 'Active data set', 'Variables' og eventuelt 'Class' og 'Output data set' (det sidste kun hvis man vil gemme resultaterne på en datafil). Vi mangler nu kun at angive, hvilke statistiske mål, vi vil have printet ud - muse-klik i de relevante bokse, højreklik på musen, vælg 'locals' og 'run' (eller brug menurækken øverst i skærmbilledet).

Hvis man er interesseret i at få printet medianen ud også, er man nødt til at gå ind i en hel anden procedure. Fra 'Data analysis' vælges 'Utilities' - 'Compute percentiles'. Derpå vælges 'Active data set', 'Variables' og 'Percentiles'. I Percentiles-vinduet vælges 50-percentilen (medianen) og evt. andre.

Der kan i øvrigt ikke beregnes Student's t test fra ASSIST.

Proc univariate:
I programeditoren kan man meget nemt via én og samme procedure få alle ovennævnte statistiske mål plus en række andre (bl.a. sandsynlighed for en normalfordelt population) samt forskellige plots ud, nemlig ved *univariate* proceduren. En opstilling med standard-valg plus test for normalfordeling ser således ud for de to variabler fra før (samme kan ikke gøres i ASSIST):

---

18 Selve programstumpen til pågældende t-test ser ud som følger:

```
Proc ttest data=ELEVER;
var height weight;
class SEX;
run;
```

```
*Programeksempel 4.3;
proc univariate data=SKOLE.ELEVER normal;
   var HEIGHT WEIGHT;
run;
```

Det vil være for omfattende at komme ind på alle de statistiske mål, der bliver beregnet her. Der henvises til "SAS Procedures Guide", kapitlet om *proc univariate*.

4.2. Frekvenstabeller

Den måde at analysere enkeltvariabler på, som de fleste vil få mest brug for, er udprintnining af univariate frekvenstabeller - *proc freq*. Herigennem får man et output-billede med tabeller af de valgte variablers fordeling på de forskellige værdier, som disse kan antage.[19]

Hvis vi f.eks. vil have en oversigt over fordelingen på variablerne for køn og vægt, 'SEX' og 'WEIGHT', samt for en variabel, som indeholder svarene på et spørgsmål om tilfredsheden med skolens lokaler og inventar, 'V01', skrives og køres følgende programlinier:

```
*Programeksempel 4.4;
proc freq data=SKOLE.ELEVER;
   tables SEX WEIGHT V01;
run;
```

For køn ser vi herefter følgende tabel i output-vinduet:

SEX	Frequency	Percent	Cumulative Frequency	Cumulative Percent
D	187	51.0	187	51.0
P	180	49.0	367	100.0

Tabellen skulle ikke give de store tolknings-problemer. I første kolonne fra venstre fremgår variabelnavnet og de to værdier, 'D' for drenge og 'P' for piger, som variablen antager. I anden kolonne ses frekvenserne, dvs. det nominelle antal drenge og piger i stikprøven. Der er altså 187 drenge og 180 piger. I tredje kolonne ses den procentdel, som de to køn hver for sig udgør. I fjerde og femte kolonne summeres henholdsvis frekvens og procent op til det samlede antal i stikprøven fra oven og nedefter - 367 elever i alt, lig med 100 pct.

[19] I forbindelse med udprintning af univariate frekvenstabeller kan der foretages Chi-square test for lige store andele i de forskellige kategorier/værdier i variablen. Denne gennemgås imidlertid i kapitel 7, hvor Chi-square test for uafhængighed mellem to variabler behandles.

Der er imidlertid et potentielt problem ved køns-variablen. ‘D’ og ‘P’ er ikke bare nogle tegn, der bliver vist i output. Variablen er *alfanumerisk* (kan antage alle cifre og tegn) og den antager simpelthen værdierne ‘D’ og ‘P’. Dette er uheldigt i forbindelse med en række analyser, hvor der forudsættes rent numeriske variabler, og jeg vil i det følgende kapitel 5 vise, hvad der kan gøres ved dette problem.

For variablen ‘WEIGHT’ printes en alenlang tabel ud, fordi elevernes vægt antager næsten samtlige værdier fra 31 til 90 (kilo). Af og til kan man have brug for at få en sådan tabel printet ud, men som oftest vil det give et langt bedre overblik over fordelingen, hvis variablen er rekodet til at indeholde interval-angivelser i stedet, og jeg vil i det følgende kapitel også komme ind på løsningen af dette problem. Med hensyn til analyse af *sammenhænge* mellem variabler skal det dog pointeres, at det kun i nogle tilfælde - f.eks. i forbindelse med krydstabeller - vil være en fordel at rekode til intervaller, mens det i andre så absolut vil være en fordel at benytte sig af den oprindelige variabel - f.eks. i forbindelse med korrelationskoefficienter og regressionsanalyse.

For variablen ‘V01’ får vi vist følgende tabel:

Lokaler og inventar

V01	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	68	18.6	68	18.6
2	106	29.0	174	47.7
3	51	14.0	225	61.6
4	81	22.2	306	83.8
5	58	15.9	364	99.7
6	1	0.3	365	100.0

Frequency Missing = 2

Bemærk først at SAS under tabellen angiver, at to observationer har ‘missing values’ i denne variabel - se mere angående manglende værdier i det følgende programeksempel. Læg dernæst mærke til, at det ikke umiddelbart er til at gennemskue, hvad de enkelte værdier fra ét til seks står for. Man er selvfølgelig normalt i besiddelse af en såkaldt *kodebog*, hvori bl.a. fremgår de nøjagtige spørgsmålsformuleringer samt værdiernes betydning. I det viste eksempel lyder spørgsmålet: *Hvor tilfreds er du med skolens lokaler og inventar?* Parantetisk kan det nævnes, at ‘inventar’ helt klart ikke er et godt ord at benytte i forbindelse med spørgsmål til skoleelever, men den slags problemer ser jeg bort fra her. Af kodebogen fremgår det endvidere, at værdierne ét til seks har følgende betydning:

1 = Meget tilfreds
2 = Noget tilfreds
3 = Hverken tilfreds eller utilfreds
4 = Noget utilfreds
5 = Meget utilfreds
6 = Ved ikke

Men at disse ting fremgår af kodebogen, er ikke ensbetydende med, at det så er nemt at overskue. *Nogle* mener, at det er fuldt tilstrækkeligt med de rene tal og gider ikke bruge tid på at gøre mere ud af det, mens andre har det langt bedre med at få printet forklarende tekst ud i stedet for. Dette gøres ved at danne og tilknytte formater, og jeg vender som nævnt tilbage hertil i følgende kapitel.

Hvis vi skal kigge lidt fremad i analyseprocessen, og det er netop et af formålene med den univariate analysedel, så er der imidlertid et andet og mere alvorligt problem med variablen 'V01'. Flertallet af de mest gængse former for analyse af sammenhænge mellem to eller flere variabler kræver et måleniveau på minimum ordinal-skala, og 'Ved ikke'-kategorien kan ikke som de øvrige rangordnes, og i hvert fald kan den ikke rangordnes på den viste måde. Det giver f.eks. ingen mening at sige, at 'Ved ikke' er større end 'Meget utilfreds', eller for den sags skyld at 'Ved ikke' skulle være udtryk for en større utilfredshed. Dette problem vil jeg komme nærmere ind på i det følgende kapitel.

En sidste ting skal her nævnes i forbindelse med frekvens-tabellen over variablen 'V01'. Vi ser, at der under tabellen angives, at der for to af eleverne i stikprøven mangler besvarelse. Når der er så få manglende besvarelser, kan det ikke siges at udgøre et problem. Havde disse derimod udgjort f.eks. 10-20 pct. af samtlige, kunne man overveje at inddrage dem i selve analysen. Dette gøres ved at indsætte en *missing-option* i programlinjerne, som vist herunder:

```
*Programeksempel 4.5;
proc freq data=SKOLE.ELEVER;
   tables V01 / missing;
run;
```

Det giver følgende output:

Lokaler og inventar

V01	Frequency	Percent	Cumulative Frequency	Cumulative Percent
.	2	0.5	2	0.5
1	68	18.5	70	19.1
2	106	28.9	176	48.0
3	51	13.9	227	61.9
4	81	22.1	308	83.9
5	58	15.8	366	99.7
6	1	0.3	367	100.0

Vi ser her, at der ud for de to *missing values* står et punktum. Hvis der ingen værdi står anført ved en *numerisk* variabel, så sætter SAS automatisk et punktum i den, hvilket er vigtigt at vide i forbindelse med de senere rekodninger. Læg mærke til, hvordan procent-angivelserne for de enkelte værdier ændres en lille smule i forhold til den sidst udprintede tabel. Ved en stor andel *missing values* vil disse værdier ændre sig betydeligt og derved give et helt andet indtryk. Som nævnt vil en beslutning om at medtage de manglende besvarelser i analysen afhænge af disses andel ud af samtlige respondenter, men derudover vil beslutningen også afhænge af, hvorvidt man antager, at de manglende besvarelser fordeler sig tilfældigt på en række - for den samlede analyse - væsentlige variabler. Noget sådan kan nemlig skævvride resultaterne og give anledning til direkte forkerte konklusioner. En anden ting er i øvrigt, at i analyser, hvor der inddrages en række variabler, som hver især måske kun har nogle få procent *missing values*, der vil resultaterne kunne komme til at hvile på et ret spinkelt grundlag - "mange bække små, gør en stor å". Dette taler ligeledes for at inddrage *missing values* i analyserne. Indragelse af *missing values* kan i øvrigt sammenlignes med problematikken af 'Ved ikke'-svarene, for heller ikke her er der tale om nogen naturlig rangorden. I kapitel 5 herunder tager jeg alle ovennævnte problemstillinger op.

***Hvordan man gør i ASSIST (univariate frekvenstabeller)*:**
Vælg 'Primary menu' - 'DATA ANALYSIS' - 'ELEMENTARY' - 'Frequency tables' - 'Generate one-way frequency tables'. Vælg herefter 'Active data set' og 'Analysis variables'. Højreklik nu på musen, og vælg 'locals' og 'run' i *pull down*-menuen (eller benyt i stedet menurækken øverst i skærmbilledet). Hvis man vil have *missing values* med i analysen, skal der dog først klikkes på 'Aditional options' - 'Customize output'. Derpå klikkes i boksen 'Include missing values in calculations of statistics' - 'OK' og til slut 'Go back'.

4.3. Hvad fortæller fordelinger i stikprøven om populationen?

Ofte vil de være af størst interesse at konkludere vedrørende sammenhænge mellem flere variabler, men af og til vil det også være interessant at gøre udsigelser om populationen blot på baggrund af *univariate* fordelinger. I den første frekvenstabel over variablen vedrørende tilfredsheden med lokaler og inventar i afsnit 4.2 så vi, at 22,2 pct. var noget utilfredse, og at 15,9 pct. var meget utilfredse (koderne 4 og 5). Altså i alt var 38,1 pct. utilfredse i en eller anden udstrækning. Men hvad siger dette statistisk set om populationen, hvis vi forudsætter, at stikprøven er taget med simpelt tilfældigt udtræk? Ja, vi må gå ud fra, at proportionen af utilfredse ligger omkring de 38,1 pct., men hvor meget kan tallet svinge?

Det forholder sig således, at tages der en lang række stikprøver, så vil deres proportion (P) - af utilfredse (0,381) eller hvad der nu er tale om - fluktuere med en normalfordeling omkring populationens proportion (π) med en standardfejl på:

$$\sqrt{\frac{\pi(1-\pi)}{N}}$$

hvor N er lig med antallet af observationer i stikprøven, da der er tale om en binomialfordeling. Nu kender vi jo ikke π, men ved at udskifte denne størrelse med stikprøve-proportionen P, gør vi kun en lille fejl, der heldigvis går mod nul ved stigende stikprøvestørrelse, og ved stikprøver på over 100 har fejlen i praksis ingen betydning, hvis P ikke ligger for langt fra 0,5. Da vi ved, at 95 pct. at observationerne i en normalfordeling falder inden for plus/minus 1,96 standardfejl, vil populationens andel af utilfredse med 95 pct. sikkerhed ligge inden for følgende interval:

$$\pi = P \pm 1{,}96\sqrt{\frac{P(1-P)}{N}}$$

P er i dette tilfælde lig med 0,381 (de 38,1 pct. i omskrevet form), og N er lig med 365 (antallet af elever). Indsættes disse tal, får vi således, at andelen af utilfredse i populationen med 95 pct. sikkerhed vil ligge inden for intervallet 33,1 pct. til 43,1 pct.

Vi kan også opstille nulhypoteser angående populations-proportionen og derpå sammenligne denne med stikprøve-proportionen. Vil vi f.eks. vide, hvor stor sandsynligheden er for, at andelen af utilfredse i populationen er på 50 pct. eller derover, så udregner vi blot en Z-værdi, hvor vi bruger den estimerede standardfejl fra før, og efterfølgende slår vi op i tabellen over normalfordelinger, hvor vi finder sandsynligheden. Den generelle formel ser således ud:

$$Z = \frac{P - H_0}{\sqrt{\frac{P(1-P)}{N}}}$$

hvor H_0 står for nulhypotese, som i dette tilfælde er sat til 0,50. Ved udregning får vi en Z-værdi på -4,68. Og et opslag i tabellen over normalfordelingen forsikrer os om, at sandsynligheden for at andel utilfredse i populationen er 50 pct. eller derover i praksis er lig nul (minus-tegnet fortæller blot, at vi udregner sandsynlighed i venstre side af normalfordelingskurven, men da kurven er symmetrisk, kan vi se bort fra fortegn).

5. REKODNING AF VARIABLER SAMT DANNELSE OG TILKNYTNING AF FORMATER OG LABELS.

Som det fremgik af kapitel 4 gennem udskrivning af univariate frekvenstabeller, var der forskellige problemer med variablerne i forbindelse med den efterfølgende analyse, og jeg vil i dette kapitel foreslå løsninger herpå. Det drejer sig om at konvertere alfanumeriske variabler til numeriske, interval-inddele variabler på interval- og ratioskala-niveau samt endelig at gøre noget ved problemet vedrørende *missing values* og 'ved ikke'-kategorien i ordinalskala-variabler. Endvidere så vi, at det kunne være vanskeligt at holde styr på, hvad de enkelte cifferkoder i tabellerne betød. Dette problem vil vi også få løst ved at danne og tilknytte såkaldte formater. Til slut i kapitlet vil jeg behandle problemer i forbindelse med såkaldte *Multiple Choice*-spørgsmål, hvortil der ofte kræves specielle rekodninger. Multiple Choice-spørgsmål er komplicerede i forhold til almindelige survey-spørgsmål, derved at der, som betegnelsen fortæller, kan gives flere svar på samme spørgsmål. Inden jeg forklarer disse forskellige former for rekodninger, vil jeg dog lige vise, hvordan man tager *back-up* af det originale datasæt og på den måde sikrer sig, at man ikke uforvarende ødelægger originale data.

5.1. Back-up af originalt datasæt

Det er altid fornuftigt at tage *back up* af det originale datasæt, inden man begynder at manipulere med det. Ligeledes kan man med fordel tage *back up* af nogle få af de senere versioner. Dette kan gøres ved blot at kopiere datasættet fra stifinderen og indsætte det et andet sted, og evt. ændre navnet så man ved, at der er tale om en back up af originalen. Et SAS-datasæt står anført i stifinderen med 'efternavnet' *sd2*. Sættet i nærværende eksempel har altså følgende fysiske betegnelse (med den fulde sti): 'h:\data\elever.sd2' (husk at navnet 'skole' blot var en henvisning til biblioteket 'data'.

Alternativt kan man kopiere datasættet i SAS, f.eks. som i nedenstående programsætninger (indledningsvist skal der selvfølgelig oprettes et bibliotek med navn 'backup'):

```
*Programeksempel 6.1;
libname BACKUP 'H:\BACKUP';
data BACKUP.ORIGINAL;
set SKOLE.ELEVER;
run;
```

Ved fremtidige kørsler i SAS behøver man ikke assigne biblioteket ‘backup’ - kun i tilfælde af, at noget er gået helt galt, og der derfor er behov for at hente backup’en af originalen. Således sikret er man klar til rekodninger. De tre sætninger - data-, set- og run-sætningen - udgør et såkaldt data-step (se Kapitel 3). Som oftest benyttes et data-step til data-manipulationer som f.eks. rekodninger, men her ses det i den simplest tænkelige form, hvor der blot sker en kopiering. I set-sætningen fortæller vi, hvilket data-sæt der benyttes som input-datasæt, og i data-sætningen fortæller vi, hvad output-datasættet skal hedde. Hvis output-sættet findes i forvejen, bliver dataene heri overskrevet, mens input-sættes bevares, som det er. I det følgende vises, hvordan der indsættes programsætninger før run-sætningen, sådan at output-sættet er forskelligt fra input-sættet.

5.2. Fra alfanumerisk til numerisk

I forbindelse med programeksempel 5.1 skrev jeg, at det ville være fornuftigt at rekode variablen for køn, ‘SEX’, fra at være en alfanumerisk til at være en rent numerisk variabel, således at bogstaverne ‘D’ og ‘P’ blev konverteret til tallene ‘0’ og ‘1’ eksempelvis. Dette kan gøres ved ganske simpelt at køre følgende lille program:

```
*Programeksempel 5.2;
data SKOLE.ELEVER2;
set SKOLE.ELEVER;
NEWSEX=0;
if SEX='D' then NEWSEX=1;
run;
```

Læg først mærke til, at output-datasættet gemmes under et nyt navn, ‘ELEVER2’, men stadigvæk under samme bibliotek, ‘SKOLE’, som input-datasættet. Selvom jeg har taget back-up af originalen, er det altid klogt at forsøge at holde sit oprindelige datasæt uberørt. Derfor danner jeg et nyt sæt i den første rekodning. I de følgende rekodninger bruger jeg sættet ‘ELEVER2’ som både input og outputsæt. Men det kan faktisk anbefales at ‘prøvekøre’ hver eneste rekodning med et midlertidigt datasæt og så checke, om de berørte variabler ser ud, som de bør. Dette gøres ved at benytte SAS-biblioteket ‘work’. Det vil sige skrive ‘work.elever’ f.eks. eller blot ‘elever’, da work-biblioteket bruges, hvis intet andet er anført. ‘Work’ er ligesom ‘library’ et reserveret ord i SAS, og det er en reference til et temporært/midlertidigt SAS-bibliotek - dvs. et bibliotek, hvis indhold slettes, så snart man igen forlader SAS. Der dannes altså et nyt datasæt, liggende under

det temporære bibliotek. Når man har checket, at rekodningerne er foretaget korrekt, kan man efterfølgende lagre datasættet permanent - enten ved at overskrive det gamle eller ved at danne et nyt. Dette kan gøres ved at udføre det samme program igen, blot hvor datasætningen er ændret.

Som det ses, er der kun to egentlige rekodnings-programlinjer i program-eksempel 5.2. I den første (NEWSEX=0;) dannes en ny variabel med navnet 'NEWSEX', og denne variabel sættes i alle observationer lig med værdien '0'. I den følgende *if*-sætning, sættes 'NEWSEX' lig med '1', hvis og kun hvis den oprindelige kønsvariabel, 'SEX', er lig med 'D'. Nu er det ikke sådan, at SAS løber alle observationer igennem linje for linje. Derimod løbes alle linjerne igennem observation for observation. Hvis den første observation/respondent/række f.eks. er en dreng, så sker der følgende: En ny variabel dannes og kommer til at ligge i slutningen af datasættet; denne variabel tilskrives værdien '0'; derpå ændres værdien til '1' pga. *if*-sætningen; og SAS går videre til den næste observation og så fremdeles. Nye variabler kan kaldes, hvad det skal være, *blot må de højst være otte karakterer lange*.

Hvis der havde været *missing values* i variablen 'SEX', ville programeksemplet være ukorrekt. Så ville nemlig de respondenter, hvortil der var *missing values*, få tilskrevet værdien '0' i den nye variabel 'NEWSEX'. For altid at være på den sikre side er det derfor klogt under alle omstændigheder at tage højde for muligheden af *missing values*. Programeksempel 5.3 viser, hvordan dette kan gøres.

```
*Programeksempel 5.3;
data SKOLE.ELEVER2;
set SKOLE.ELEVER;
if SEX='P' then NEWSEX=0;
else if SEX='D' then NEWSEX=1;
else NEWSEX=.;
run;
```

Her spørges specifikt til begge køn gennem en *if then else*-sætning, og programordenes betydning kan oversættes direkte til dansk: hvis - så - ellers. Logikken i sætningen er den, at først spørger man, om det er en pige. Hvis det er en pige, så får 'NEWSEX' værdien '0'. Ellers (dvs. hvis ikke) spørger man, om det er en dreng. Hvis det er tilfældet, får 'NEWSEX' værdien '1', og hvis det hverken er en pige eller en dreng, får 'NEWSEX' værdien '.'. Som det fremgik af outputet til programeksempel 5.2, så betyder punktum i SAS-sprog *missing value*, når der er tale om numeriske variable. Så når vi tilskriver en variabel værdien punktum, er det i virkeligheden det samme som at fortælle, at der ikke skal stå noget i den. Det skal lige nævnes her, at hvis man spørger direkte til *missing valmues* i en *alfanumerisk* variabel (variabler der kan antage værdier i form af både tal,

bogstaver og andre tegn), eller hvis man vil sætte "værdien" af en *alfanumerisk* variabel til *missing*, så skal der benyttes notationen *anførselstegn-blanktegn-anførselstegn* (' ') i stedet for punktum.

I ovennævnte eksempel kunne jeg i øvrigt helt udelade den sidste sætning (`else NEWSEX=.;`). Numeriske felter (variabler i observationer), der ikke eksplicit tildeles en værdi, sættes nemlig *automatisk* lig med et punktum. Det giver dog en ekstra sikkerhed at have linjen med. Hvis f.eks. input- og output datasæt er det samme (hvilket det ganske vist ikke er i dette tilfælde), og man tidligere har lavet en fejl-rekodning, så er de felter, der burde være tomme, måske tildelt en reel værdi, som "overlever" den nye rekodning. Men om man gør det ene eller det andet afhænger meget af, hvordan sikkerheden indbygges i de vaner, man får i sin programmering.

Efter rekodning ser en frekvenstabel over den nye variabel, 'NEWSEX', således ud[20]:

NEWSEX	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	180	49.0	180	49.0
1	187	51.0	367	100.0

5.3. Dannelse og tilknytning af format samt påsætning af label.

Efter rekodningen af kønsvariablen har vi det problem, at det kan være svært at huske, hvilket tal der står for hvilket køn. Jeg vil derfor danne et format, som kan knyttes til den nye variabel, således at der i stedet for '0' og '1' bliver skrevet 'Piger' og 'Drenge'. Som nævnt er dette ikke noget, der får indflydelse på analyserne, men kun på output (SAS bruger de bagvedliggende tal i beregningerne). Jeg danner formatet således:

```
*Programeksempel 5.4;
proc format library=library;
value   FKON     0='Piger'
                 1='Drenge'
                 ;
run;
```

Der skrives det lidt ulogiske 'library=library', fordi formatet skal lagres i format-*biblioteket*, og dette bibliotek har jo netop det faste kaldenavn 'library'. Man kan spørge, hvorfor det så ikke er lavet sådan, at man slet ikke behøver skrive noget, men det er der i hvert fald én god grund til. Hvis man ikke skriver 'library=library', altså bare 'proc

[20] I forhold til den tidligere viste tabel over variablen for køn figurerer drenge og piger nu i omvendt rækkefølge. SAS skriver dem fra den mindste til den største værdi, oppefra og ned, og da 'P', som er større end 'D', har fået værdien '0', som er mindre end '1', så vil det være sådan.

format;', så dannes der kun et midlertidigt format i workbiblioteket, der, som tidligere nævnt, slettes, når SAS lukkes ned.

Ordet 'value' skal altid anføres i forbindelse med denne type formater, mens 'FKON' er et selvvalgt navn for det format, der skal knyttes til den nye kønsvariabel. Bogstavet 'F' i starten af navnet er blot for at nemme forståelsen; man kan kalde formatet hvad som helst - blot maksimalt otte tegn, og til forskel fra variabelnavne må det ikke slutte med et nummer. Semikolonnet, der står lige under et-tallet, kunne lige så godt være placeret umiddelbart efter det sidste anførselstegn. Det er en vanesag, hvordan man foretrækker sådanne layout-mæssige detaljer. På samme vis med størrelsen af tabulerings-indryk eller blanktegn rundt omkring.

Ekskurs: Om permanente kontra midlertidige formater

I forbindelse med format-biblioteker er det væsentligt at bemærke, at det ofte kan være en rigtig god ide udelukkende at benytte sig af formater, der er gemt i det midlertidige work-bibliotek og så blot gemme programmerne, der laver formaterne. Alene det at skulle holde styr på alle de formater, man i tidens løb får hobet op, kan være et stort problem. Ydermere vil man ofte skulle analysere på datasæt, som andre har siddet med først, og man giver selv datasæt videre til andre. Fordelen ved de permanente formater er selvfølgelig, at de kan genbruges i efterfølgende projekter, men det kræver et gevaldigt godt overblik og en god ordenssans. Yderligere fordele ved at benytte work-biblioteket (ud over at det er nemmere at holde styr på, og man ikke risikerer at overskrive formater, der ikke skulle overskrives) er, at man ikke behøver 'assigne' formatbibliotek, og at de nemt kan ændres, hvis der skulle opstå behov herfor.

Selvom der altså gennemgående i denne vejledning vises eksempler med formater, der gemmes permanent, vil det derfor for mange være anbefalelsesværdigt at holde sig til formater i work-biblioteket. Meget vigtigt er det imidlertid, at man i så fald husker at gemme *programmerne*, der danner de midlertidige formater. Vælger man imidlertid at benytte et permanent bibliotek til sine formater, kan det være en stor fordel at få udskrevet oplysninger om samtlige formater på et givent bibliotek. Dette gøres ved hjælp af en "format"-procedure. Eksempelvis udskrives oplysninger om samtlige formater på biblioteket "library" med følgende procedure.

```
Programeksempel 5.5
proc format library=library fmtlib;
run;
```

For formatet "FKON", som jeg lige har dannet, ser output ud som vist herunder, og det fremgår, at vi bl.a. kan se, hvilken værdi der hører til hvilket køn. Det kan synes lidt overflødigt, at der både angives start- og slutværdi på kategorierne, men det hænger sammen med, at man kan lave formater, hvor hver kategori dækker over et interval (se herom senere):

```
„ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ†
‚       FORMAT NAME: FKON     LENGTH:    6   NUMBER OF VALUES:    2       ‚
‚   MIN LENGTH:   1  MAX LENGTH:  40  DEFAULT LENGTH   6  FUZZ: STD       ‚
‡ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ…ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ…ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ‰
‚START           ‚END             ‚LABEL  (VER. 6.12    15MAR00:10:23:17)  ‚
‡ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ‰
‚               0‚               0‚Piger                                   ‚
‚               1‚               1‚Drenge                                  ‚
Šƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ‹ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ‹ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŒ
```

◆ ◆ ◆

Efter denne lille ekskurs vender jeg tilbage til det konkrete eksempel. Jeg har nu oprettet et køns-*format* og skal derpå knytte dette til den nye køns-*variabel*, og samtidig vil jeg påhæfte en såkaldt *label*. En label er en kort forklaring/betegnelse af variablen. Variabelnavnet må jo maksimalt fylde otte karakterer, og da et survey-datasæt ofte vil indeholde langt over 50 variabler, vil det efterhånden komme til at knibe med at huske, hvad de enkelte variabelnavne står for[21]. Ved at påsætte labels, kan man få forklaringer ud i output. Format og label tilknyttes således:

```
* Programeksempel 5.6;
data SKOLE.ELEVER2;
set SKOLE.ELEVER2;
format NEWSEX FKON.;
label NEWSEX='Køn - rekodet til numerisk';
run;
```

Format-sætningen knytter formatet 'FKON' til variablen 'NEWSEX'. *Det er meget vigtigt, at der er et punktum umiddelbart efter formatnavnet*, og her må der ikke være nogen mellemrum. Hvis ikke der sættes et punktum, tror SAS at der er tale om et nyt variabelnavn. Man kan nemlig udmærket knytte et og samme format til flere variabler. Man skriver blot variabelnavnene efter hinanden efterfulgt til sidst af formatnavnet plus punktum. Derefter kan sætningen afsluttes med semikolon, eller der kan fortsættes med andre formattilknytninger.

[21] Et godt råd i forbindelse med navngivning af variabler, når man selv foretager en spørgeskemaundersøgelse, er at opkalde dem efter spørgsmålsnumrene, således f.eks. at variablen til spørgsmål nummer 7 får navnet 'V07', og hvis der er flere underspørgsmål, kan de f.eks. kaldes 'V07A' og 'V07B'. 'V' står for 'variabel', men der kan benyttes et

Format- og labeltilknytningen ovenfor samt rekodningen, som jeg lavede i afsnit 5.1., kan selvfølgelig laves i ét skridt. Det ser således ud:

```
* Programeksempel 5.7;
data SKOLE.ELEVER2;
set SKOLE.ELEVER;
if SEX='P' then NEWSEX=0;
else if SEX='D' then NEWSEX=1;
else NEWSEX=.;
format NEWSEX FKON.;
label NEWSEX='Køn - rekodet til numerisk';
run;
```

Og en frekvensudskrift af den formaterede kønsvariabel ser ud som følger:

'Køn - rekodet til numerisk'

NEWSEX	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Piger	180	49.0	180	49.0
Drenge	187	51.0	367	100.0

hvilket som helst bogstav - variabelnavnet skal blot begynde med et bogstav (og hvis variablerne skal bruges som elementer i tabeller, skal de ende på fortløbende cifre, ikke bogstaver).

5.4. Inddeling i intervaller (fra interval- til ordinalskala) samt mere om formater.

I kapitel 4 skrev jeg, at variablen 'WEIGHT' i nogle situationer var uoverskuelig at arbejde med. Den kan simpelthen antage for mange værdier, og tabeludskrifter vil ikke kunne overskues. Ved eventuelle krydstabeller, hvor der testes for sammenhænge mellem variabler, vil det gå helt galt, og en statistisk test som Chi-square vil være ubrugelig. Jeg inddeler derfor denne variabel samt også 'HEIGHT' og 'AGE' i et tilpas antal intervaller, således at de rekodede variabler antager værdier svarende til disse intervaller. Hvis jeg vælger at inddele i tre intervaller, vil variablerne altså kun kunne antage tre forskellige værdier, f.eks. '1', '2' og '3'[22]. Jeg vil først danne formaterne for derefter at foretage rekodningen og tilknytte de dannede formater samt også tilføje labels. Dette kan eksempelvis se ud som i programeksempel 5.8 herunder. Et godt råd, når der skal køres flere programstumper, er at køre dem hver for sig, så man bedre kan overskue eventuelle fejlmeddelelser. Altså markér først programmet til formatdannelsen (*proc format)* og kør det. Hvis det går godt, så markér derefter rekodningsprogrammet (*data-step'et*) og kør det.

```
* Programeksempel 5.8;
proc format library=library;
value   FVGT     1='Let'
                 2='Middel'
                 3='Tung'
                 ;
value   FHOJ     1='Lav'
                 2='Middel'
                 3='Høj'
                 ;
value   FALD     1='Ung'
                 2='Ældre'
                 3='Ældst'
                 ;
run;
```

[22] Det er normalt i en sådan situation at begynde med værdien '1', hvorimod det ved såkaldte binære variabler eller dummy-variabler, som de også kaldes, er normalt at benytte sig af værdierne '0' og '1'. Selvom det i mange analyser ikke betyder noget, hvilke koder man benytter, blot at værdierne i ordinalskala-variabler rangordnes, og at man kender denne rangorden, så vil det være fornuftigt alligevel at gøre ovennævnte til en norm.

```
data SKOLE.ELEVER2;
set SKOLE.ELEVER2;

if WEIGHT<=.z then NEWWGHT=.;
else if WEIGHT<=50 then NEWWGHT=1;
else if WEIGHT<=65 then NEWWGHT=2;
else NEWWGHT=3;

if HEIGHT<=.z then NEWHGHT=.;
else if HEIGHT<=150 then NEWHGHT=1;
else if HEIGHT<=170 then NEWHGHT=2;
else NEWHGHT=3;

if AGE<=.z then NEWAGE=.;
else if AGE<12 then NEWAGE=1;
else if AGE<15 then NEWAGE=2;
else NEWAGE=3;

format NEWWGHT FVGT. NEWHGHT FHOJ. NEWAGE FALD.;
label   NEWWGHT='Vægt - tre kategorier'
        NEWHGHT='Højde - tre kategorier'
        NEWAGE='Alder - tre kategorier'
        ;
run;
```

For det første lægger vi mærke til tegnsammensætningen '<=’. Dette betyder mindre end eller lig med, og inden jeg forklarer programmet nøjere, viser jeg herunder de forskellige typer af *sammenlignings-operatorer*, som findes i SAS:

Sammenlignings-operatorer i SAS

Symbol	*Såkaldt mnemonisk ækvivalent**	*Definition*
=	EQ	Lig med (equal to)
^=	NE	Ikke lig med (not equal to)**
>	GT	Større end (greater than)
<	LT	Mindre end (less than)
>=	GE	Større end eller lig med (greater than or equal to)
<=	LE	Mindre end eller lig med (less than or equal to)
	IN	Lig med én i en liste (equal to one of a list)***

* Disse kan benyttes i stedet for symbolerne.

** 'Ikke lig med' *kan* være anderledes, afhængigt af computer-systemet.

*** Der findes intet tegnsymbol til denne operator

I programeksempel 5.8 har jeg - for at vise forskellige muligheder - sat if then else-sætningerne op på en lidt anden måde end ved rekodningen af kønsvariablen. Her spørges direkte til, om den pågældende variabel er blank, f.eks. i sætningen: '`if WEIGHT<=.z then`

NEWWGHT=.;‘. Læg mærke til, at jeg ikke nøjes med at spørge, om variablen er *lig med* et punktum, men derimod om den er *lig med eller mindre end* ‘.z’. Som oftest vil man kunne nøjes med at spørge, om den er lig med et punktum, men i nogle tilfælde vil det gå galt. I SAS er der nemlig mulighed for at definere forskellige typer af *missing values* med bogstaverne ‘a’ til ‘z’ samt *underscore*, ‘_’. Så vidt jeg ved, er det ikke meget udbredt i surveydata, men man *kan* komme ud for det. Underscore er defineret som værende mindst, dernæst følger et alenestående punktum, så ‘.a’ til ‘.z’, hvorefter de valide numeriske værdier følger - først de negative, så nul og til slut de positive.

Hvis man føler sig lidt usikker på if then else-sætninger, kan rekodningen i stedet skrives med rene if-sætninger - f.eks. således for variablen ‘AGE’ (jeg laver et midlertidigt datasæt, da der kun er tale om en prøve):

```
* Programeksempel 5.9;
data work.ELEVER;
set SKOLE.ELEVER2;
if WEIGHT<=.z then NEWWGHT=.;
if 0<WEIGHT<=50 then NEWWGHT=1;
if 50<WEIGHT<=65 then NEWWGHT=2;
if WEIGHT>65 then NEWWGHT=3;
format NEWAGE FALD.;
run;
```

Jeg spørger her i hver sætning, om variablens værdi ligger inden for et interval - f.eks. i sætningen ‘`if 0<WEIGHT<=50 then NEWWGHT=1;`’. Hvis ‘WEIGHT’ er større end ‘0’ og mindre end eller lig med ‘50’, så skal ‘NEWWGHT’ sættes lig med ‘1’. For fuldstændighedens skyld, gives herunder endnu et programeksempel, der laver nøjagtigt den samme rekodning. Her vises brugen af de *logiske operatorer* ‘and’, ‘or’ og ‘not’, som i øvrigt også kan skrives med tegnene ‘&’, ‘|’ og ‘^’. Hvis man skriver ‘and’, betyder det, at *både* det ene og det andet skal være sandt. Hvis man skriver ‘or’, betyder det, at *én af delene eller begge dele* skal være sand(e).

```
* Programeksempel 5.10;
data work.ELEVER;
set skole.ELEVER2;
if WEIGHT<=.z then NEWWGHT=.;
if WEIGHT>=0 and WEIGHT<=50 then NEWWGHT=1;
if WEIGHT>50 and WEIGHT<=65 then NEWWGHT=2;
if WEIGHT>65 then NEWWGHT=3;
run;
```

En frekvenstabel for den rekodede variabel for vægt, 'NEWWGHT', ser således ud:

Vægt - tre kategorier

NEWWGHT	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Let	109	29.8	109	29.8
Middel	155	42.3	264	72.1
Tung	102	27.9	366	100.0

Frequency Missing = 1

Hvis man skulle komme til at knytte et forkert format til en variabel, knytter man bare det rigtige til på samme vis, så erstatter dette den gamle tilknytning. Og hvis man kommer til at knytte et format til en variabel, der slet ikke skulle knyttes noget brugerdefineret format til, så fjerner man blot tilknytningen. Dette sidste kan gøres ved at lave en tom format-sætning, som i programeksempel 5.11 for variablen 'newage':

```
* Programeksempel 5.11;
data work.ELEVER;
set work.ELEVER;
format NEWAGE;
run;
```

Der findes endnu en metode til sammenlægning af kategorier/værdier. Det kan anbefales, at man til en start bruger den ovenfor beskrevne metode, men jeg viser herunder i programeksempel 5.12 et alternativ - igen for variablen 'WEIGHT'. Jeg starter med at danne et format, hvor intervallerne afgrænses, og derefter tilknyttes dette format den oprindelige variabel. Frekvenstabeller og krydstabeller samt tilhørende Chi-square test vil benytte sig af de tre intervaller, mens en række andre analyser, f.eks. regressionsanalyse, vil benytte de bagvedliggende ikke formaterede værdier.

```
* Programeksempel 5.12;
proc format library=library;
value   F2VGT    low-50='Let'
                 50<-65='Middel'
                 65<-high='Tung'
                 ;
run;

data ELEVER;
set SKOLE.ELEVER2;
WEIGHT2=WEIGHT;
format WEIGHT2 F2VGT.;
run;
```

Intervallet '`low-50`' betyder fra og med 0 til og med 50. 'Low' inkluderer altså ikke *missing values*. Intervallet '`50<-65`' betyder over 50 til og med 65, og '`65<-high`' betyder over 65 til maksimum værdi. Mindre end-tegnet signalerer altså, at den pågældende værdi

ikke skal med, og det kan fint sættes på den anden side af bindestregen, alt afhængig af hvordan intervallerne præcist skal ordnes. I data step-programmet starter jeg med at kopiere vægt-variablen over i '`WEIGHT2`' og knytter derefter formatet til denne nye variabel. Jeg kunne udmærket have knyttet formatet til den oprindelige, men så ville jeg afskære os for nogle analysemuligheder. Man kan nemlig komme ud for at have brug for de oprindelige tal i en SAS-funktion, hvor der kun vises de formaterede værdier. Hvis uheldet er ude, kan man dog altid fjerne det, sådan som det blev vist i programeksempel 5.11.

5.5. 'Missing values' og 'Ved ikke'-kategorier.

Som det blev nævnt i Kapitel 4 er mange variabler i f.eks. survey-data på ordinalskala-niveau, men som oftest vil der i disse variabler også være kategorier, der ikke naturligt passer ind i rangordenen. Dette gør sig gældende for variablerne '`v01`' til '`v11`'. Disse variabler indeholder svarene på en serie tilfredshedsspørgsmål og antager værdierne '1' til '5', som naturligt kan rangordnes, fordi de signallerer en større eller mindre tilfredshed. Men variablerne kan også antage værdien '6', som står for 'Ved ikke', og der kan desuden forekomme *missing values*.

Én måde at løse problemet på, er at transformere værdier, der ikke naturligt passer ind i rangordenen, til midtersvaret. En anden er at sætte dem til *missing values*. Jeg vælger her at transformere 'Ved ikke'-besvarelserne til midterkategorien ('Hverken tilfreds eller utilfreds'), hvilket jo *på sin vis* også er korrekt, idet vedkommende jo netop hverken er det ene eller det andet. På den anden side er der en årsag til, at man ofte i konstruktionen af spørgeskemaet har givet mulighed for begge svarmuligheder, så helt synonyme er de to altså ikke oprindeligt tænkt.

Ofte vil en eksplicit angivelse af 'Ved ikke' signallere manglende viden, mens angivelse af 'Hverken tilfreds eller utilfreds' signallerer noget, man kunne kalde mellemfornøjethed, hvor respondenten mener at have tilstrækkeligt med viden til at kunne foretage en vurdering, men hvor vedkommende vurderer det, der spørges til, som værende hverken tilfredsstillende eller utilfredsstillende. Og af den grund er en sådan rekodning af 'Ved ikke'-besvarelser altid noget, der skal ekspliceres, således at læseren selv kan tage stilling til lødigheden.

Missing values lader jeg være, men man kunne også vælge at transformere disse om til midterkategorien, hvilket ganske vist ville være endnu mere diskutabelt. I praksis er det dog ofte, men langt fra altid, ret ubetydelige forskelle, de forskellige metoder giver i den videre analyse. Vær imidlertid opmærksom på, om der kun er få 'Ved ikke'-besvarelser og *missing values*, og (hvis antallet er stort eller det ligger på grænsen)

undersøg dernæst om respondenterne med disse besvarelser/værdier er tilfældigt fordelt på vigtige baggrundsvariabler. Hvis ikke dette er tilfældet, bør metoden genovervejes. I programeksempel 5.13 viser jeg rekodningen (’Ved ikke’ til midterkategori), inklusiv formatdannelse.

```
* Programeksempel 5.13;
proc format library=library;
value   FTILFR   1='Meget tilfreds'
                 2='Noget tilfreds'
                 3='Hverken eller'
                 4='Noget utilfreds'
                 5='Meget utilfreds'
                 6='Ved ikke'
                 ;
run;


data SKOLE.ELEVER2;
set SKOLE.ELEVER2;
if V01=6 then REKV1=3;
else REKV1=V01;
if V02=6 then REKV2=3;
else REKV2=V02;
if V03=6 then REKV3=3;
else REKV3=V03;

<Rekodning af V04-V10 er ikke vist>

if V11=6 then REKV11=3;
else REKV11=V11;

format V01--V11 REKV1--REKV11 FTILFR.;

run;
```

Det eneste nye i programmet, hvad angår det rent programeringsmæssige, er den måde, hvorpå jeg har tilknyttet formatet til variabler. De to bindestreger mellem variabelnavnene angiver, at format-tilknytningen også skal gælde for alle de variabler, der i datamatricen ligger mellem de nævnte. Det er en måde at liste variabler på, som også kan bruges i andre situationer, f.eks. i forbindelse med SAS-funktioner, som bl.a. beskrives i afsnit 8.1. Jeg viser herunder forskellige former for variabel-lister i SAS.

Forskellige typer af variabel-lister i SAS (eksempler med forklaring)

Eksempler	*Variabler som er med i listen*
`var1--var4`	De to nævnte plus alle variabler der i data-matricen ligger mellem disse.
`var1-numeric-var4`	Som ovenstående, blot kun gældende for de numeriske variabler.
`var1-character-var4`	Som ovenstående, blot kun gældende for karakter-variablerne.
`var1-var4`	Alle variabler med samme 'karakternavn' (her 'var'), som slutter med cifrene 1, 2, 3, og 4.
`_numeric_`	Samtlige numeriske variabler i data-sættet.
`_character_`	Samtlige karakter-variabler i data-sættet.
`_all_`	Samtlige variabler i data-sættet.

Program-eksempel 5.13 er forholdsvist langt, og noget er da også udeladt, fordi rekodningen af de enkelte variabler ligner hinanden. Ved flittig brug af klippe-klistre metode kan man dog hurtigt fremstille lange programmer, så ofte vil det ikke være så slemt, som det ser ud på papiret - man skal blot huske at 'holde tungen lige i munden'. Hvis man imidlertid vil gøre det lidt simplere (i hvert fald at se på), kan den samme rekodning laves med tabeller (i computerterminologi ofte nævnt med den engelske term *arrays*), som vist herunder (uden formatdannelsen):

```
* Programeksempel 5.14;
data ELEVER2;
set SKOLE.ELEVER2;

array TAB_TF(11) V01-V11;
array REKV(11);
do I=1 to 11;
        if TAB_TF(I)=6 then REKV(I)=3;
        else REKV(I)=TAB_TF(I);
end;

format V01--V11 REKV1--REKV11 FTILFR.;
drop I;

run;
```

Den første *array*-sætning sætter de eksisterende variabler 'V01' til 'V11' til at udgøre en éndimensionel tabel, og den anden *array*-sætning danner nye variabler med navnene 'REKV1' til 'REKV11', som udgør en tilsvarende tabel. *Do end*-løkken udføres 11 gange (når f.eks. 'I' er lig med 3, står 'REKV(I)' for 'REKV3'). Sidst i programmet er nu kun format-tildelingen samt sætningen 'drop I;' tilbage. Drop-sætningen sørger for, at variablen 'I' ikke gemmes i datasættet (denne skulle kun bruges midlertidigt, og det er klogt at rydde op efter sig, så der ikke efterhånden kommer til at ligge en masse

overflødige variabler i data-sættet. Der findes også en keep-sætning i SAS, som kan benyttes i situationer, hvor det er mere overskueligt og nemt at fortælle, hvilke variabler, der skal *med* i data-sættet, i stedet for hvilke variabler, der ikke skal med.

Arbejde med arrays forklares mere indgående nedenfor i afsnit 5.7. Afsnittet er lidt mere kompliceret program-teknisk set end de øvrige afsnit i vejledningen, og det kan uden problemer betragtes som en ekskurs og springes over, så fremt der ikke arbejdes med Multiple Choice-spørgsmål eller arrays. Der gennemgåes ganske vist andre nye programelementer i afsnittet, men disse gennemgåes atter i afsnit 8.1. vedrørende dannelsen af indeks. Umiddelbart herunder, i afsnit 5.6. vil vi se på sammenlægning af værdier i variabler, der i forvejen har et begrænset antal værdier - altså ordinal- eller nominalskalerede variabler.

5.6. Sammenlægning af kategorier i variabler.

Vi så i afsnit 5.5., hvordan vi kunne løse problemet med missing values og 'Ved ikke'-kategorier, således at vi fik variabler, der var brugbare i forbindelse med analysefasen. Imidlertid vil man ofte have behov for yderligere rekodning. For selv om f.eks. en ordinalskaleret variabel som 'REKV1' (rekodet fra variabel 'V01') kun kan antage et begrænset antal værdier, nemlig heltal fra '1' til '5', så vil det ofte være en fordel at sammenlægge kategorier, sådan at antallet af mulige værdier f.eks. indskrænkes fra '5' til '3'. Eksempelvis kan man komme i en situation, hvor antallet af observationer i de enkelte kategorier er for fåtallig til, at der kan udføres statistiske test, eller at man vil sammenligne to variabler, hvor den ene oprindeligt kan antage fem værdier og den anden kun tre. Principielt kan denne rekodning foretages på samme måde som transformeringen fra intervalskalerede til ordinalskalerede variabler, sådan som det blev vist i afsnit 5.4 - nemlig med 'if then else'-sætninger, som sikkert er det almindeligste . Programeksempel 5.15 herunder viser dette.

```
* Programeksempel 5.15;
data LOLLE.ELEVER2;
set LOLLE.ELEVER2;
if REKV1=1 or REKV1=2 then REKV1B=1;
else if REKV1=3 then REKV1B=2;
else if REKV1=4 or REKV1=5 then REKV1B=3;
else REKV1B=.;
run;
```

Programmet kan imidlertid gøres nemmere og især mere overskueligt. Med brug af programsætningen 'select' kan den ovenfor viste rekodning udføres på følgende måde:

```
* Programeksempel 5.16;
data SKOLE.ELEVER2;
set SKOLE.ELEVER2;
select(REKV1);
        when(1,2) REKV1B=1;
        when(3)   REKV1B=2;
        when(4,5) REKV1B=3;
        otherwise REKV1B=.;
        end;
run;
```

I 'select'-sætningen vælges en variabel, og dernæst 'evalueres' hver enkelt 'when'-sætning, indtil der i parantesen findes en værdi, der er lig med den valgte variabels værdi. Findes der en 'when'-sætning med den korrekte værdi, udføres den efterfølgende 'ordre'. Altså hvis eksempelvis variablen 'REKV1' er lig med '4', så sættes den nye variabel 'REKV1B' til værdien '3'. Hvis ikke der findes en 'when'-sætning med den korrekte værdi, så udføres 'otherwise'-sætningen, som i øvrigt ikke må udelades. Denne metode virker ofte langt mere overskuelig, og især gælder dette, når alternativet ville være en lang række 'if then else'-sætninger.[23]

5.7. Multiple Choice-spørgsmål

Et såkaldt Multiple Choice-spørgsmål er et spørgsmål, hvortil det er tilladt respondenten at afgive flere svar. Der vil som regel være opstillet en række faste svarkategorier i tilknytning til spørgsmålet, hvor man så kan afkrydse en eller flere; og her skal man i konstruktionen af spørgeskemaet sørge for meget tydeligt at forklare, at der kan afgives flere svar samt evt. at notere et maksimalt antal svar. Denne type spørgsmål kan være behagelige at medtage i et spørgeskema (og nogle gange nødvendige), men analysen er til gengæld mere kompliceret end ved almindelige spørgsmål med ét svar. Og allerede i indkodningen bør man gøre sig nogle tanker om, hvordan svarerne skal bruges i analysen. Principielt to forskellige metoder at indtaste på skal kort omtales her.

Mest almindeligt - og sikkert også nemmest at arbejde videre med - er det at lade hver svarkategori optræde som selvstændig variabel, som kan antage to værdier. Sådanne variabler, der kan antage to, og kun to, værdier, kaldes ofte for *dummy*-variabler eller *dikotome* variabler. Værdierne vil her normalt være '0', som står for uafkrydset, og '1', som står for afkrydset.

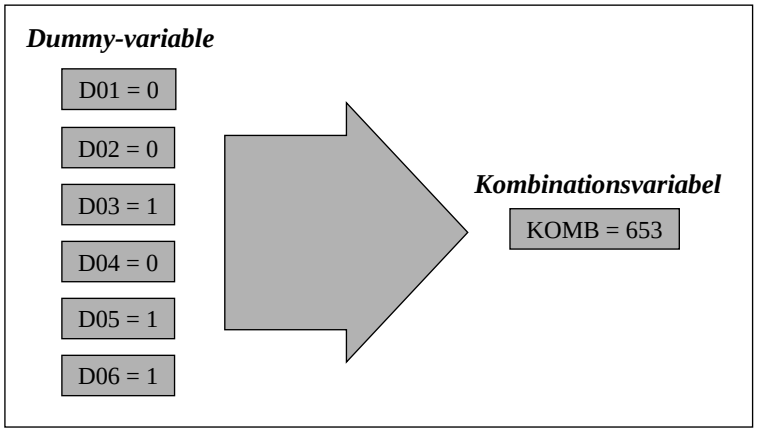
Et andet alternativt er at kode alle svarene i én og samme variabel. Hvis f.eks. respondenten har seks valgmuligheder og har afkrydset nr. 3, 5 og 6, så indtastes blot tallet 356 eller 653. Der er både ulemper og fordele ved begge de nævnte metoder.

[23] Programeksempel 19 kan imidlertid også gøres noget mere overskueligt på anden vis - nemlig ved brug af sammenligningsoperatoren 'in' (se afsnit 9.1. for et eksempel på brug heraf).

Ved førstnævnte er det umiddelbart muligt at analysere på svarkategorierne hver for sig, men ikke den samlede svarkombination, mens det forholder sig lige omvendt ved den sidstnævnte metode.

Det anbefales at benytte førstnævnte metode, hvor analysen af enkelte svarkategorier er lige til. Uanset hvilken metode, man vælger at indtaste efter, viser det sig dog ofte, at man alligevel gerne have mulighed for at analysere på den anden måde også - altså både på enkeltkategorier og på den samlede svarkombination. Jeg viser derfor herunder, hvordan transformation af variabler kan udføres for begge veje. Først vises, hvordan en serie dummy-variabler kan transformeres om til en enkelt variabel indeholdende den eksakte kombination. Hvis vi genkalder os førnævnte eksempel med besvarelse af kategori nr. 3, 5 og 6, så vil følgende transformering skulle foretages:

Transformering af dummy-variabler fra Multiple Choice-spørgsmål til kombinationsvariabel.



Hver af variablerne D01 til D06 er en dummy-variabel, der kan antage værdien '0' eller '1', hvor '1' signallerer, at svarkategorien er afkrydset. Hvis spørgsmålet slet ikke er besvaret, har alle variabler *missing values*. Spørgsmålet drejer sig om, hvilke fag eleverne gerne vil arbejde med på et højere niveau. '`D01`' står f.eks. for dansk, '`D02`' for matematik og '`D03`' for engelsk. I tilknytning til spørgsmålet er det noteret, at der højst må afkrydses fire fag.

Transformeringen kan foretages via et forholdsvis nemt lille program, som vises herunder. Der sker dog nogle nye ting rent programmeringsmæssigt, hvorfor jeg efterfølgende vil forklare programmet indgående.

```
* Programeksempel 5.17;
data SKOLE.ELEVER2;
set SKOLE.ELEVER2;

ARRAY TABEL(6) D01-D06;
TAELLER=0;
KOMB=0;

if sum(of D01-D06)>4 or D01=. then KOMB=.;
else do I=1 to 6;
        if TABEL(I)=1 then do;
                KOMB=KOMB+I*(10**TAELLER);
                TAELLER=TAELLER+1;
        end;
end;

drop TAELLER I;
run;
```

Foruden arrays benyttes i programeksemplet to nye ting, nemlig funktioner og aritmetiske operatorer. Herunder vises, hvordan tegnene til aritmetiske operatorer ser ud i SAS, og de er stillet op i den rækkefølge, som de udregnes i, hvor dog plus og minus er ligestillede (normal matematisk rækkefølge). Hvis man vil ændre rækkefølgen, skal der indsættes paranteser, ligesom det kendes fra almindelige matematik.

Aritmetiske operatorer i SAS

Symbol	*Definition*
**	Exponential
*	Multiplikation
/	Division
+	Addition
-	Subtraktion

Hvis eksempelvis variablen ‘taeller’ er lig med ‘2’, vil ‘10**TAELLER’ være lig med ‘10^2’, og hvis vi f.eks. vil lægge to variables (`D01` og `D02`) værdier sammen og multiplicere denne sum med to, kan dette skrives som ‘`(D01+D02)*2`’.

I den første if-sætning i programeksemplet er der indsat en sum-funktion, ‘`sum(of D01-D06)`’. Denne sammentæller værdierne i variablerne ‘`D01`’ til ‘`D06`’. Læg i øvrigt mærke til, at der i angivelsen af variabel-listen kun er benyttet én bindestreg, hvor jeg tidligere har benyttet to. Herved fortælles, at listen omfatter de nævnte to variabler plus de variabler, der har samme bogstav-navn, og som har afsluttende cifre liggende mellem de nævnte, og udfyldende hele intervallet – den går altså ikke, hvis der f.eks. ikke findes nogen variabel med navnet ‘`D03`’. Til gengæld er der ikke krav om, at variablerne

skal ligge placeret fysisk ved siden af hinanden i datamatricen, sådan som det er tilfældet ved benyttelse af to bindestreger (reglerne for variabel-lister er beskrevet i afsnit 5.5). Det er meget vigtigt at huske det foranstillede 'of' til listen. SUM-funktionen kan nemlig indeholde både lister, enkeltvariabler og aritmetiske udtryk, og hvis der alene står '`sum(D01-D06)`', tror SAS, at der er tale om et aritmetisk udtryk – dvs. summen af den ene variabel trukket fra den anden, hvilket netop giver den ene variabel trukket fra den anden (det er selvfølgelig ulogisk efter dagligdagsfornuft, når der kun er ét aritmetisk udtryk, men for konsistens i reglerne er det logisk). Der kunne istedet være skrevet: '`sum(D01, D02, D03, D04, D05, D06)`'. Det ville give samme resultat.

Foruden sum-funktionen findes i SAS en lang række funktioner, hvoraf jeg lister nogle få herunder.

Eksempler på funktioner i SAS. Funktionsnavnet efterfølges i programmet af en parentes indeholdende enten et argument* eller en liste med variable/værdier. Ofte vil funktionen indgå i en forbindelse som følgende:

VARIABELNAVN=FUNKTIONSNAVN(ARGUMENT ELLER VARIABEL-LISTE);

Funktion	Beskrivelse (funktionen returnerer)
ABS	Absolut værdi af argument
MAX	Maksimum værdi blandt de listede
MIN	Minimum værdi blandt de listede
SIGN	Fortegn: -1 hvis mindre end 0; 0 hvis 0; 1 hvis større end 0
N	Antal valide værdier blandt de listede
NMISS	Antal *missing values* blandt de listede
SUM	Sum af de listede
CEIL	Mindste heltal større end eller lig med argument
FLOOR	Største heltal mindre end eller lig med argument
INT	Heltalsværdi af argument
ROUND	Afrundet værdi** af argument

* Et argument kan være enten et variabelnavn, en konstant eller et matematisk udtryk. Og et matematisk udtryk kan indeholdende variabelnavne, konstanter eller begge dele samt en eller flere aritmetiske operatorer.

** Hvis der f.eks. skrives 'round(121.453)' vil det returnerede tal blive '121.000' (husk at bruge punktum i stedet for komma). Man kan imidlertid afrunde på selvvalgt decimal i stedet. F.eks. vil 'round(121.453,.1)' returnere værdien '121.500' og 'round(121.453,.01) vil returnere værdien '121.450'.

Mange funktioner laver de samme regneoperationer, som man vil kunne klare med de aritmetiske operatorer. Der er imidlertid en meget afgørende forskel, nemlig i hvordan der tages hensyn til *missing values*. Hvis man benytter aritmetiske operatorer, så vil

resultatvariablen blive sat til *missing*, hvis blot én af variablerne i det matematiske udtryk er *missing*. Ved brug af funktioner vil dette ikke ske. I ovennævnte programeksempel gør det ingen forskel, om man skriver alle variabel-navnene med plus-tegn imellem, eller om man benytter sum-funktionen, da det under indtastningen er sørget for, at hvis en af variablerne har *missing value*, så har de øvrige det også. Men i mange tilfælde kan det få afgørende betydning, om man benytter funktioner eller aritmetiske operatorer, og i afsnit 8.1. er dette eksemplificeret. I det følgende vil jeg imidlertid koncentrere mig om, hvad der helt præcist foregår i programeksempel 5.17.

I sætningen '`array TABEL(6) D01-D06`'; fortæller jeg SAS, at de seks dummy-variabler skal betragtes som en tabel (med navn `TABEL`), der kan refereres til med et indeks. Derpå initieres to nye variabler, en tæller samt den nye kombinationsvariabel, til startværdien '0'. I den første if-sætning sættes kombinations-variablen til missing value, hvis der er afkrydset mere end fire fag, jeg har sat som øvre grænse, og/eller hvis dummy-variablerne har missing values (det er her nok at spørge til en enkelt, da jeg under indtastningen har sørget for enten at sætte alle til missing values (hvis der ingen valg er foretaget) eller at udfylde alle variabler med '0' eller '1')[24]. Hvis ikke kombinations-variablen skal være missing, så gennemløbes 'do end-løkken' seks gange, hvor en til lejlighed oprettet variabel '`I`' benyttes som tæller. Hvis man ikke er vant til at arbejde med tabeller, kan det være lidt vanskeligt at holde styr på, hvad der egentlig sker, så jeg gennemgår herunder de seks gennemløb minutiøst med udgangspunkt i ovennævnte eksempel, hvor en elev havde afkrydset ud for fag nr. tre, fem og seks.

I de to første gennemløb udføres slet ingen beregninger, da disse kun udføres, hvis dummy-variablen (eller tabel-elementet) er lig med '1'. I tredje gennemløb vil de to sætninger med udregninger se således ud:

```
KOMB=0+3*(10**0);
TAELLER=0+1;
```

hvorfor variablen '`KOMB`' bliver lig med '3' og variablen '`TAELLER`' bliver lig med '1'. Variablen '`TAELLER`' benyttes altså for at holde styr på, hvor mange fag der er afkrydset, og vi vil se den dybere mening med variablen i gennemløb nr. fem. Gennemløb fire paserer forbi uden beregninger som de første to, men i gennemløb fem, hvor '`D05`' ('`TABEL(5)`') er lig med '1', foretages de to beregninger igen, og sætningerne ser nu sådan her ud:

[24] Det kan diskuteres, hvorvidt ingen valg skal resultere i missing values. Det afhænger helt af, om man forventer, at mindst et af svarmulighederne bør benyttes, eller om det tillades ikke at foretage noget valg.

```
KOMB=3+5*(10**1);
TAELLER=1+1;
```

Først udregens parantesen, som egentlig er overflødig, fordi potensopløftninger som nævnt altid udføres først, men indsættelse af paranteser kan af og til hjælpe med til at gøre udregningerne mere overskuelige. Tallet '10' opløftes altså i første potens, hvilket giver '10'; derpå multipliceres dette ti-tal med '5', hvilket giver '50'; og til sidst summeres tallene '3' og '50', sådan at variablen 'KOMB' sættes lig med '53', hvilket betyder, at kategori tre og fem er afkrydset. I næste sætning tælles variablen 'TAELLER' op med én, således at den nu bliver lig med '2'. I sjette og sidste gennemløb ser de to sætninge således ud:

```
KOMB=53+6*(10**2);
TAELLER=2+1;
```

'10' opløftes i anden potens, hvorefter dette tal multipliceres med '6', hvilket giver '600'. Derpå sættes variablen 'komb' lig med '53' plus '600', hvilket giver '653', hvilket lige netop var, hvad jeg gerne ville ende op med, fordi eleven havde afkrydset ud for fag nr. tre, fem og seks. Variablen 'taeller' tælles til sidst op med én, men det er i øvrigt ligegyldigt nu, da beregningerne er færdige, og jeg springer ud af løkken. Sidst i programmet er nu kun sætningen '`drop TAELLER I;`' tilbage, som sørger for, at variablerne 'TAELLER' og 'I' ikke gemmes i datasættet.

Eksemplet ovenfor kan med små justeringer benyttes til lignende situationer med færre eller flere svarkategorier og med færre eller flere som maksimalt antal svar (evt. helt uden grænse). Imidlertid skal der en ret afgørende justering til, førend programmet kan benyttes til Multiple Choice-spørgsmål med ti eller flere svarkategorier, for der er kun afsat én ciffer-plads til hvert svar i kombinations-variablen. Eneste, men vigtige, ændring i programmet er her at tælle variablen 'TAELLER' op med to i stedet for én, hver gang der findes en ny afkrydsning. Herved skabes to cifferpladser til hvert svar. Hvis denne ændring var foretaget med det førnævnte eksempel, hvor der var afkrydset i kategori tre, fem og seks, ville variablen 'KOMB' være lig med '60503'.

Når man har beregnet sin kombinations-variabel, kan denne benyttes som enhver anden nominalskaleret variabel. Man kan dog risikere, at der findes et hav af forskellige kombinationer, hvilket vil komplicere analysen. Ofte vil man dog kunne argumentere for, at en del af kombinationerne kan slås sammen, alt efter hvad man ønsker at sige noget om med sine analyser. Til slut vil jeg ganske kort beskrive, hvordan transformeringen den modsatte vej - fra kombinationsvariabel til dummy-variabler - kan arrangeres programmæssigt.

```
* Programeksempel 5.18;
data SKOLE.ELEVER2;
set SKOLE.ELEVER2;
array D(6);
do I=1 to 6;
        D(I)=0;
end;
KOMB2=KOMB;

if KOMB ne . then do;
        do until (KOMB2=0);
                I=KOMB2-(int(KOMB2/10)*10);
                D(I)=1;
                KOMB2=int(KOMB2/10);
        end;
end;
else do I=1 to 6;
        D(I)=.;
end;

drop KOMB2 I;
run;
```

Beregningerne i dette eksempel er en tand mere komplicerede end ved transformeringen den modsatte vej, så det er med at holde koncentrationen, når programmet laves, samt efterfølgende at checke dets funktion grundigt igennem. Programmet vil ikke blive forklaret lige så detaljeret, som det foregående, men hovedtrækkene skitseres.

Der dannes først otte nye variabler, 'D1' til 'D6', i array-sætningen (læg mærke til, at en array-sætning uden efterfølgende variabel-navne fører til dannelse af nye variable). Derpå nulstilles alle otte dummy-variabler, og der dannes en kopi af kombinations-variablen 'KOMB'. Hvis kombinations-variablen ikke er lig med *missing value*, så gøres følgende lige så mange gange, som der er cifre i kombinations-variablen: Variablen 'I' sættes lig med det sidste ciffer (mod højre) i variablen 'KOMB2' (kopien); dummy-variabel nr. 'I' sættes til værdien '1'; fra variablen 'KOMB2' fjernes det sidste ciffer mod højre. Hvis kombinations-variablen *er* lig med *missing value*, så sættes alle dummy-variabler til missing value (else-sætningen). Til slut droppes variablerne 'KOMB2' og 'I'.

Hvis der er ti eller flere kategorier i Multiple Choice-spørgsmålet, og der derfor er benyttet to cifre til hver kategori i kombinations-variablen, da ændres tallet '10' alle steder i programeksemplet til '100'.

Multiple Response:

Såkaldt Multiple Response minder i kodning og analyse om Multiple Choise. Multiple Response er et åbent spørgsmål - dvs. der er blot afsat plads, evt. med indtegnede linier, til at respondenten kan notere et eller flere svar uden faste kategorier. Ved en manuel

gennemgang af en lang række besvarelse dannes et overblik over, om besvarelserne kan kategoriseres i et forholdsvis overskueligt antal kategorier. Hvis dette er tilfældet, dannes et antal variabler lig med antallet af kategorier på samme måde som ved Multiple Choise-spørgsmål. En væsentlig forskel er dog, at man ofte vil lade den første variabel lig med den førstindskrevne besvarelse i stedet for den første kategori; den anden variabel lig med nr. to indskrevne besvarelse osv. Hvis f.eks. en responden har besvaret med det, der efterfølgende er bestemt som kategori nr. 4, 1 og 6 i nævnte rækkefølge, så får variabel nr. 1 værdien 4; variabel nr. 2 får værdien 1; og variabel nr. 3 værdien 6. Dette gøres, fordi den af respondenten indskrevne rækkefølge af besvarelserne kan signallere en prioritering, således at den først indskrevne findes vigtigst. Metoden kan også benyttes i en situation, hvor der er faste svarkategorier, men hvor man beder respondenten om f.eks. at prioritere vigtigheden af svarene, således at der ikke blot krydses af i svarfelterne, men sættes et tal svarende til prioriteringen - f.eks. ‘1’ for førsteprioritet, ‘2’ for andenprioritet osv. Under alle omstændigheder kan man - for at simplificere - have interesse i at analysere svarene uden hensyntagen til evt. prioritering; og der skal herunder vises et program, der kan transformere variablerne om til dummy-variabler, hvor hver variabel står for en kategori og ikke et svarnummer.

```
* Programeksempel 5.19;
data SKOLE.ELEVER2;
set SKOLE.ELEVER2;
array D(8);
array TABEL(8) SVAR1-SVAR8;

if SVAR1 ne . then do I=1 to 8;
        D(I)=0;
end;

if SVAR1 ne . then do I=1 to 8;
        if TABEL(I) ne . then D(TABEL(I))=1;
end;

drop I;
run;
```

Variablerne ‘SVAR1’ til ‘SVAR8’ står for hvert af de mulige svar, respondenten kan give, således f.eks. at værdien af ‘SVAR1’ henviser til et kategori-nummer. Programmet er i forhold til de tidligere viste ret simpelt og skal ikke forklares nærmere. Eneste nye aspekt er, at i sidste if-sætning benyttes værdien af et tabelelement som indeks i en anden tabel.

6. BIVARIAT SAMMENHÆNG

Jeg skal nu gennemgå den bivariate analyse. I afsnit 6.1 herunder behandles først krydstabeller med tilhørende test for henholdsvis sammenhæng og ingen sammenhæng mellem to *nominalskalerede* variabler, og i afsnit 6.2 behandles mål for sammenhæng mellem to *ordinalskalerede* variabler.

Til nominalskalerede variabler kan Chi-square testen benyttes til at fortælle om sandsynligheden for *u*afhængighed i *populationen* mellem to variabler, dvs. risikoen for at begå en type to fejl, ifald man konkluderer, at der *er* sammenhæng. Chi-square testen kan imidlertid også bruges til at teste for sandsynligheden for lige store andele af de forskellige kategorier i en *univariat* fordeling, og der gives et eksempel herpå. Ud over Chi-square testen vises to mål for sammenhængs*styrke* mellem to nominalskalerede variabler, nemlig *Cramer's V* og *Goodman & Kruskal's lambda*.

Korrelations-koefficienter som *Gamma* og *Somers' d* benyttes i forbindelse med ordinalskalerede variabler (eller evt. dikotome variable), og de fortæller noget om styrke og retning i sammenhængen mellem to variabler, hvor '-1' angiver perfekt negativ sammenhæng, '0' angiver ingen ordinal sammenhæng, og '+1' angiver perfekt positiv sammenhæng. Samtidig kan sikkerheden for at sammenhængen f.eks. er forskellig fra '0' beregnes ud fra den angivne standardfejl.

6.1. Krydstabeller, Chi-square test samt mål for sammenhængsstyrke mellem to nominalskalerede variabler.

Den mest grundlæggende metode til analyse af bivariate sammenhænge er ved hjælp af krydstabeller, og det er essentielt, at man kan fortolke krydstabeller, før man begiver sig ud i mere avancerede analyser. At metoden er grundlæggende og ikke kan betegnes som avanceret, betyder i øvrigt ikke, at den så kun benyttes af begyndere i statistisk analyse. I en række situationer er krydstabeller det bedste alternativ, i andre er de en god start. Og så har de endnu den fordel, at de egner sig fortrinligt til præsentation for et bredt publikum. Vigtigt i den forbindelse er i øvrigt, at tabellerne præsenteres på en informativ og overskuelig måde og *ikke* som rene SAS-udskrifter.

Jeg vil i det følgende se på de tre bivariate sammenhænge mellem variablerne for elevernes alder, højde og vægt. Jeg benytter de rekodede variabler '`NEWAGE`', '`NEWHGHT`' og '`NEWWGHT`', fordi disse har tilpas få kategorier til at de med stikprøvestørrelsen

in mente kan give mening i tabellerne samt til Chi-square beregningerne. I program-eksempel 6.1 vises i en og samme procedure, hvordan de tre tabeller skrives ud. Procecduren er den samme som ved univariate frekvensudskrifter. De to variabler i en krydstabel skrives blot med en stjerne mellem, hvor variablen før stjernen er rækkevariabel og variablen efter stjernen er kollonnevariabel. Det kan i øvrigt være en god ide at gøre det til en vane altid at nævne den uafhængige variabel først og den afhængige til sidst, sådan at man aflæser ens krydstabeller på den samme måde fra gang til gang. Af og til kan det selvfølgelig være vanskeligt at vurdere, hvilken variabel der er det ene eller det andet, altså hvilken vej kausaliteten vender, men når det er indlysende, er det fornuftigt som hovedregel at vende variablerne i en bestemt retning. Mest almindeligt er at vende dem på ovennævnte facon - altså med den uafhængige først og derpå den afhængige.[25]

```
* Programeksempel 6.1;
proc freq data=SKOLE.ELEVER2;
   tables NEWAGE*NEWHGHT NEWAGE*NEWWGHT NEWHGHT*NEWWGHT / chisq measures;
run;
```

Udskrivning af krydstabeller over bivariate sammenhænge sker altså ved at indsætte en stjerne mellem de to variabelnavne, og med det viste eksempel får vi udskrevet tre krydstabeller. Skråstregen er indsat for at fortælle SAS, at nu følger der en fordring om bestemte statistiske mål samt ting i forbindelse med tabelindholdet. Her beder jeg om Chi-square baserede test (chisq) samt om forskellige mål for sammenhængens styrke (measures). Hvis vi vil have de univariate fordelinger ud tillige, så skriver vi blot dem på også på følgende måde (de tre variabler uden stjernetegn mellem):

```
* Programeksempel 6.2;
proc freq data=SKOLE.ELEVER2;
   tables       NEWAGE NEWHGHT NEWWGHT
                NEWAGE*(NEWHGHT NEWWGHT) NEWHGHT*NEWWGHT / chisq measures;
run;
```

Læg mærke til, at jeg med parantesen kan nøjes med at skrive '`NEWAGE`' én gang. Her er det selvfølgelig begrænset, hvor meget arbejde der er sparet, men hvis én og samme variabel skal krydses med en række variabler, kan denne metode klart betale sig. Og ydermere,

[25] Problemerne med at bestemme kausalitetsretningen angår i øvrigt også nærværende gennemgående eksempel. De fleste vil ganske vist være enige i, at alder logisk set kan betragtes som kommende før højde og vægt. Derimod kan der være tvivl om, hvordan det forholder sig med højde og vægt indbyrdes. Umiddelbart forekommer det vel mest logisk at betragte højde som førstkommende, men i virkeligheden kan det jo være svært at sige, om man vokser først i højde eller i rummål (og dermed vægt). I øvrigt kan det tilføjes angående den mere praktiske side, at passer en tabel-udskrift bedre på papiret, når den vendes omvendt den sædvanlige, kan man selvfølgelig bare gøre det alligevel, istedet for at skulle ændre sideopsætning – mere faste skal reglerne heller ikke være.

hvis det er muligt at beskrive variablerne i parantesen som en variabel-liste, er dette også tilladt (se om variabel-lister i afsnit 5.5). En del af output'et til programeksempel 6.2 vises herunder; først de univariate fordelinger til 'NEWHGHT' og 'NEWWGHT' og dernæst den bivariate sammenhæng mellem disse to variabler. Det kan være vanskeligt at bedømme, om der er tale om en *effekt* fra højde på vægt, eller om der bare er tale om sammenhæng mellem de to variabler. Jeg har her valgt at betragte højden som uafhængig og vægt som afhængig. De univariate fordelinger vises kun, fordi jeg her til forskel fra tidligere får beregnet Chi-square test for lige store andele af kategorierne i de enkelte variabler.

```
                Højde - tre kategorier

                                    Cumulative  Cumulative
NEWHGHT   Frequency   Percent   Frequency    Percent
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
Lav            120      32.7         120       32.7
Middel         114      31.1         234       63.8
Høj            133      36.2         367      100.0


        Chi-Square Test for Equal Proportions
        -------------------------------------
    Statistic = 1.542     DF = 2     Prob = 0.462


                Vægt - tre kategorier

                                    Cumulative  Cumulative
NEWWGHT   Frequency   Percent   Frequency    Percent
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
Let            109      29.8         109       29.8
Middel         155      42.3         264       72.1
Tung           102      27.9         366      100.0

                Frequency Missing = 1


        Chi-Square Test for Equal Proportions
        -------------------------------------
    Statistic = 13.590     DF = 2     Prob = 0.001
```

'Prob' angiver sandsynligheden for lige store andele i populationen, og det ses, at ved højde kan en hypotese herom ikke afvises, mens sandsynligheden for lige store andele af de tre vægtkategorier er højst usandsynlig. Det beregningstekniske i Chi-square testen vises nedenfor i forbindelse med test for uafhængighed mellem to variabler.

TABLE OF NEWHGHT BY NEWWGHT

NEWHGHT(Højde - tre kategorier) NEWWGHT(Vægt - tre kategorier) Frequency, Percent , Row Pct , Col Pct	Let	Middel	Tung	Total
Lav	80	39	0	119
	21.86	10.66	0.00	32.51
	67.23	32.77	0.00	
	73.39	25.16	0.00	
Middel	29	70	15	114
	7.92	19.13	4.10	31.15
	25.44	61.40	13.16	
	26.61	45.16	14.71	
Høj	0	46	87	133
	0.00	12.57	23.77	36.34
	0.00	34.59	65.41	
	0.00	29.68	85.29	
Total	109	155	102	366
	29.78	42.35	27.87	100.00

Frequency Missing = 1

I feltet for oven til venstre i tabellen angives det, hvad tallene i de enkelte celler betyder. F.eks. at det øverste tal i hver celle er frekvensen, dvs. det nominelle antal observationer (elever i dette tilfælde). Hvis man har vendt variablene, som det blev anbefalet med den afhængige til sidst, så vil det først og fremmest være rækkeprocenterne (altså tredje talrække i hver kategori for 'NEWHGHT'), man skal koncentrere sig om. Man 'scanner' ned over tabellen og ser, hvorvidt der sker ændringer i mønstret for rækkeprocenterne. I dette tilfælde er der markante ændringer, således at jo højere eleven er, des større vægt er der tendens til. F.eks. er ca. to tredjedele (67,23 pct.) af de *lave* elever *lette*, mens ca. to tredjedele (65,41 pct.) af de *høje* elever er *tunge*. Man kan billedligt udtrykke det sådan, at der er en bevægelse diagonalt fra øverste venstre hjørne til nederste højre hjørne. Når jeg bevæger os nedad gennem tabellen, flyttes tyngdepunktet mod højre.

Nedenstående Chi-square test bekræfter indtrykket fra tabelaflæsningen - se forklaring efterfølgende. De statistiske mål er delt i to blokke. Øverst findes de Chi-square baserede mål, som alle kan benyttes til nominalskalerede variabler. I dette afsnit beskrives selve *Chi-square testen*, som vises i øverste linie, samt *Cramer's V*. Nederst findes en blok med forskellige andre mål, der tester for sammenhængens styrke, og nogle også for retning. Vær meget opmærksom på, at nogle af disse mål kræver interval/ratio-skalerede variabler, andre kræver mindst ordinalskalerede variabler, og atter andre egner sig til de nominalt skalerede (SAS regner alle målene ud, så længe der blot ligger talkoder bag variablenes kategorier, og man skal derfor selv sørge for at holde styr på, hvilke der må benyttes at tolke på, og hvilke der ikke må benyttes). I dette afsnit ser jeg fra denne

blok på *Somers' D* samt *lambda-koefficienten*, da begge disse er konstrueret til nominalskalerede variabler.

STATISTICS FOR TABLE OF NEWHGHT BY NEWWGHT

Statistic	DF	Value	Prob
Chi-Square	4	219.889	0.001
Likelihood Ratio Chi-Square	4	260.459	0.001
Mantel-Haenszel Chi-Square	1	192.590	0.001
Phi Coefficient		0.775	
Contingency Coefficient		0.613	
Cramer's V		0.548	

Statistic	Value	ASE
Gamma	0.890	0.019
Kendall's Tau-b	0.669	0.024
Stuart's Tau-c	0.662	0.026
Somers' D C|R	0.663	0.026
Somers' D R|C	0.674	0.023
Pearson Correlation	0.726	0.023
Spearman Correlation	0.727	0.023
Lambda Asymmetric C|R	0.389	0.059
Lambda Asymmetric R|C	0.446	0.045
Lambda Symmetric	0.419	0.048
Uncertainty Coefficient C|R	0.329	0.023
Uncertainty Coefficient R|C	0.325	0.024
Uncertainty Coefficient Symmetric	0.327	0.024

Effective Sample Size = 366
Frequency Missing = 1

Chi-square:

Chi-square benyttes som nævnt ovenfor til at teste en H_0 hypotese om uafhængighed mellem tabellens to variabler. Resultatet opgives som sandsynligheden for uafhængighed, og jeg vil herunder kort redegøre for, hvordan denne sandsynlighed udregnes.

I udregningen af 'Value' for Chi square benyttes antal forventede i hver celle i krydstabellen, under forudsætning om *ingen* sammenhæng i populationen. Det forventede antal i hver enkelt celle findes ud fra følgende formel:

$$E = \frac{rækketotal \times kolonnetotal}{N}$$

hvor *E* er lig med antal forventede observationer i den enkelte celle, *N* er lig med antal observationer i alt i stikprøven, og række- og kolonnetatal er det nominelle antal observationer i den respektive henholdsvis række og kolonne. Eksempelvis vil det forventede antal observationer i øverste venstre celle i ovenfor viste krydstabel altså skulle findes således:

$$E = \frac{119 \times 109}{366} \approx 35$$

Og formlen for Chi-square value er herefter:

$$\chi^2 = \frac{(O-E)^2}{E}$$

hvor O er lig med det observerede antal observationer i den enkelte celle. Den heraf fundne værdi bruges sammen med antallet af frihedsgrader: $(k-1)\times(r-1)$, hvor *k* er lig antal kolonner, og *r* er lig antal rækker, til at finde en p-værdi fra Chi-square distributionen (findes bagerst i de fleste statistikbøger)[26]. Den nøjagtige p-værdi udregnes imidlertid også af SAS og angives under 'Prob' (probability of no association). I ovenfor viste udskrift er p-værdien højst 0,001, og det betyder, at risikoen for at konkludere forkert ved at påstulere sammenhæng i populationen, er én promille eller derunder. Man siger også, at *signifikansniveauet* er mindre end eller lig med 0,001.

I mange tilfælde står man i en situation, hvor ens krydstabel viser en fin sammenhæng, men hvor der under Chi-square testens output angives en *warning* om, at så og så stor en andel celler i tabellen har en forventet frekvens på under fem, eller at en eller flere rækker eller kolonner er helt tomme. SAS giver en advarsel, hvis andel celler med under 5 forventede observationer er på 20 pct. eller derover. Nogle er af den mening, at dette er en lidt lav grænse, men kommer andelen over 30 pct. er der i hvert fald for stor tvivl om, hvorvidt man uden videre kan stole på Chi-square testen.

I nogle tilfælde kan problemet løses ved at rekode variablene, sådan at flere kategorier slås sammen til én, hvorved der kommer flere observationer i de enkelte celler. Eksempler på en sådan rekodning er vist i afsnit 5.4. I andre situationer kan det forsvares, at man løser problemet ved analyse på såkaldte *subset*, hvor en eller flere variabel-kategorier helt trækkes ud i analysen - typisk 'Ved ikke'-kategorier, hvor der ofte kun vil være en forholdsvis lille frekvens. Hvis ingen af disse muligheder står åbne, kan man gå ind i en nøjere undersøgelse af, hvorvidt man på trods af advarslen måske alligevel kan stole på Chi-square testen. Disse to sidstnævnte muligheder (altså analyse på subset samt den nøjere undersøgelse af Chi-square testens validitet) gives der eksempler på i afsnit 7.1.

[26] Formlen for Chi-square value for test for lige store andele i univariat fordeling er den samme. Det forventede antal er her blot lig med stikprøvens samlede antal observationer divideret med antallet af kategorier, og antallet af frihedsgrader er lig med antallet af kategorier minus én.

Cramer's V:

Chi-square testen fortæller ikke direkte noget om sammenhængens *styrke*. På basis af Chi-square testen er der imidlertid udviklet forskellige mål, der kan sige noget herom. Pearson, som intruducerde Chi-square testen udviklede efterfølgende *Phi* og *Contingency* koefficienterne, og Cramer intruducerede senere koefficienten *Cramer's V*, som er en variant af disse. Fordelen ved denne sidstnævnte er, at den altid befinder sig mellem '0' (ingen sammenhæng) og '+1' (perfekt sammenhæng). Formlen for Cramer's V er:

$$Cramer's\ V = \sqrt{\frac{\chi^2}{N(k-1)}}$$

hvor *N* er lig med antal observationer, og *k* er lig med det mindste antal rækker/kollonner (hvis der f.eks. er tre rækker og fem kollonner, er *k* lig med tre). I ovennævnte eksempel er Cramer's V lig med 0,538, hvilket er en ret stærk sammenhæng[27]

Lambda:

Cramer's V kan måske være vanskelig begribelig som udtryk for styrke i sammenhæng. Intuitivt er det sikkert nemmere at forstå de mål for sammenhæng mellem to nominalskalerede variabler, der benytter *proportional reduction in error* (PRE). *Lambda* er et sådant mål, og det tager udgangspunkt i den procentandel fejl, man laver, hvis man prøver at forudsige udfaldet på den afhængige variabel, dels når man ingen kendskab har om den uafhængige variabel, dels når man har et sådant kendskab.

Jeg vil igen se på det ovenfor viste eksempel med sammenhængen mellem '`NEWHGHT`' og '`NEWWGHT`'. Hvis vi *ikke* har kendskab til værdien på den uafhængige vil det mest kvalificerede gæt på udfaldet af '`NEWWGHT`', være 'Middel', da der befinder sig flest elever i denne kategori - nemlig 42,35 pct. (se under krydstabellen). Fejlprocenten er derfor 57,65 (100-42,35). Hvis vi derimod *har* kendskab til værdien på den uafhængige, reduceres procentandelen af fejlgæt betragteligt: 32,77 pct. for 'Lav', 38,6 (100-61,40) for 'Middel' og 34,59 for 'Høj'. Herefter tages et vejet gennemsnit af disse tre fejlandele:

$$\frac{(32{,}77\times119)+(38{,}60\times114)+(34{,}59\times133)}{366}\approx 35{,}25$$

[27] Der er ikke nogen fastsat regel for, hvor stor værdi Cramer's V skal have, før sammenhængen kan betragtes som værende stærk. Og ofte vil værdien da også have mest interesse i forbindelse med sammenligning mellem forskellige sammenhænge og ikke som et absolut mål for styrke.

Reduktionen i andel fejl er altså ca. 22,4 (57,65-35,25) procent-*point*, og den proportionelle eller forholdsvise reduktion - altså Lambda - er derfor:

$$Lambda\ Asymmetric\ C|R = \frac{P(1) - P(2)}{P(1)} \approx \frac{57{,}65 - 35{,}25}{57{,}65} \approx 0{,}389$$

som det også står anført i den ovenfor viste statistikudskrift. *P*(1) er fejlproportion *uden* kendskab til værdien af den uafhængige variabel, og *P*(2) er fejlproportion *med* kendskab til denne. Målet er asymmetrisk, fordi størrelsen afhænger af, hvilken variabel, der betragtes som afhængig. Med 'C|R' menes således, at kollonnevariablen (NEWWGHT) er den afhængige og rækkevariablen (NEWHGHT) er den uafhængige. Der angives i udskriften også en omvendt samt en symmetrisk version af lambda-koefficienten. Vigtigt i forbindelse med Lambda er, at den er et mål for en bestemt type af sammenhæng, nemlig *fejlreduktion*. Der kan forekomme situationer, hvor Lambda er lig med '0', men hvor andre statistiske mål faktisk viser en sammenhæng. Som det er anført i en manual: "A measure of association sensitive to every imaginable type of association does not exist" (Norusis 1990, p. 125).

Vi har nu set, at mens Chi-square testen fortæller om sikkerhed/signifikans, fortæller de øvrige to nævnte statistiske mål, Cramer's V og Lambda, noget om sammenhængens styrke. Vi får imidlertid ingenting at vide om, hvorvidt der findes en bestemt orden i denne sammenhæng, sådan at stigende værdi på den uafhængige giver tendens til enten stigende eller faldende værdi på den afhængige. En sammenhæng mellem to variabler kan jo udmærket skifte fra side til side, og når vi taler om nominalskalerede variabler, er vi netop også kun interesseret i sådanne ikke nærmere specificerede mål, fordi der ingen rangorden er i nominalskalerede variables værdier. I det ovenfor viste eksempel er begge variabler imidlertid på ordinalskala niveau, og jeg har en hypotese om, at stigende højde giver tendens til stigende vægt. Derfor vil jeg gerne have statistiske mål for den *rangordensmæssige* sammenhæng, og ikke blot en hvilken som helst form for sammenhæng. Fra krydstabellen kunne vi med det blotte øje se, at der var en tendens til stigning på den afhængige variabel ('NEWWGHT'), når den uafhængige ('NEWHGHT') steg, men dels vil man ofte komme ud for tvivlstilfælde, dels vil man gerne have et standardiseret mål for den rangordensmæssige sammenhængs styrke og retning. Sådanne mål kan fås via beregning af korrelations-koefficienter som f.eks. Gamma, Somers' d og Kendall's tau b, og jeg vil gennemgå disse i det følgende kapitel.

***Hvordan man gør i ASSIST (bivariate sammenhæng, krydstabel)*:**
Vælg 'Primary menu' - 'DATA ANALYSIS' - 'ELEMENTARY' - 'Frequency tables' - 'Generate n-way crosstabulation table'. Vælg herefter 'Active data set' og 'Analysis variables'. Der er nu blot angivet, hvilke variabler der skal med i analysen - ikke hvilke tabeller der skal printes ud. Klik på knappen 'crosstabulations' og vælg nu de relevante tabeller (krydstabellerne står angivet som variabelnavne med stjerne imellem ligesom i programmet). Højreklik nu på musen, og vælg 'locals' og 'run' i *pull down*-menuen (eller benyt i stedet menurækken øverst i skærmbilledet).

Hvis man vil have Chi-square test og eller korrelationskoefficienter beregnet, skal man dog først klikke på 'Additional options' - 'Statistics'. Chi-square testen fås ved at sætte flueben i den øverste boks, 'calculate test of no association', og *Gamma*-koefficienten fås ved at sætte flueben i nummer to boks, 'calculate measures of association'. Derpå klikkes 'OK' - 'Go back' og analysen kan køres med 'locals' og 'run'.

Undertrykkelse af talstørrelser i tabelceller samt krav om yderligere talstørrelser vælges under 'Additional options' - 'customize output' (Nogle tal kan ikke "bestilles" i ASSIST). Og dannelse af sub-set, hvor kategorier undertrykkes, foretages ved at klikke på 'Subset data' - 'WHERE clause', hvorefter der med en del museklik og evt. lidt skriveri bygges en *where clause* op. Husk at gå ind og trykke 'reset' - 'ok', når I igen vil analysere på det samlede datasæt.

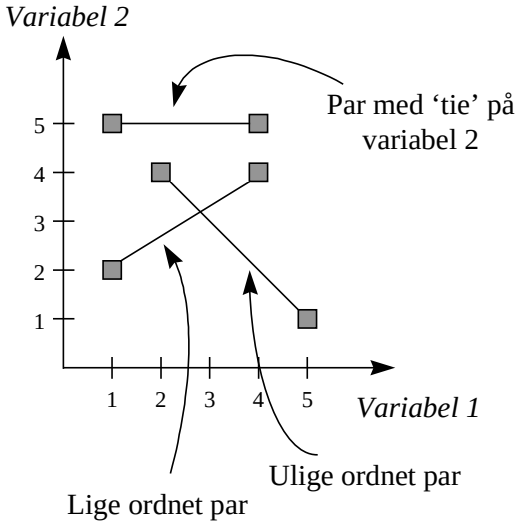
6.2. Korrelation

Vi fandt i afsnit 6.1 i krydstabel med tilhørende Chi-square test samt andre statistiske mål en overbevisende sammenhæng mellem 'HEIGHT' og 'WEIGHT', men vi fandt ikke noget statistisk belæg for at konkludere, at denne sammenhæng har en bestemt retning, samt heller ikke hvor stærk denne sammenhæng egentlig er. Hertil kan, hvis variablerne er på mindst ordinalskala-niveau, benyttes korrelationskoefficienter. Disse fortæller om retning, styrke og sikkerhed og går fra '-1' (perfekt negativ sammenhæng) over '0' (ingen rangordensmæssig sammenhæng) til '+1' (perfekt positiv sammenhæng). Chi-square testen kan udmærket være stærkt signifikant, uden at en korrelationskoefficient er det - nemlig i det tilfælde hvor der er sammenhæng, men at denne skifter fra side til side ned over krydstabellen. Jeg vil hovedsageligt i dette kapitel behandle koefficienterne *Goodman og Kruskal's Gamma* og *Somer's d*, som (desværre) kun kan beregnes og udskrives i forbindelse med krydstabeller. Jeg vil dog kort omtale andre, og i kapitel 7 vil jeg behandle partielle korrelationskoefficienter.

Gamma:

Gamma-koefficienten er en såkaldt rang-korelationskoefficient, som sammenligner observationerne parvist og optæller antal par, som er ordnet lige (konkordante), samt par som er ordnet ulige (diskordante) med hensyn til de to variabler.

Et lige ordnet par vil sige, at hvis der på den ene variabel er en stigning i værdi fra den ene observation i parret til den anden, så vil der også være en stigning i værdi på den anden variabel. Modsat vil et ulige ordnet par sige, at hvis der på den ene variabel er en stigning i værdi fra den ene observation i parret til den anden, så vil der være en *faldende* værdi på den anden variabel. Observationer, hvor der ikke kan foretages sammenligning, fordi parrenes observationer har samme værdi i den afhængige eller uafhængige variabel eller begge dele (såkaldte *ties*), lades ude af betragtning.



Anmærkning: Par med 'tie' på variabel 1 ville ses som en lodret streg.

Optællingerne af ulige og lige ordnede par benyttes til at beregne en Gamma-koefficient, der ligger mellem -1 og 1, og formlen er som følger:

$$Gamma = \frac{P - Q}{P + Q}$$

hvor *P* er antal lige ordnede par, og *Q* er lig med antal ulige ordnede par. Værdien angiver andelen af overvægt af konkordante par over diskordante, blandt par der ikke har tie på nogen af variablerne. Af formlen ses tydeligt, at: er der lige mange lige og ulige ordnede par, da vil Gamma være '0'; er der kun lige ordnede par, vil Gamma være '+1'; og er der kun ulige ordnede par, vil Gamma være '-1'.

Det kan diskuteres, hvorvidt det er hensigtsmæssigt, at par med ties ikke indregnes i formlen for Gamma-koefficienten, men et faktum er, at den herved har tendens til at blive større end de fleste andre korrelationskoefficienter, hvilket da også tydeligt fremgår af output-eksemplet herunder.

Somers' d:

En *Asymmetrisk Somers' d* indregner ties på den afhængige variabel, som ikke samtidigt har ties på den uafhængige. I figuren ovenfor er netop vist et par med tie på den afhængige variabel, og det burde være logisk, at hvis der forekommer forholdsvis mange par, hvor værdierne på den uafhængige variabel (effekt-variablen, hvis der er tale om kausalitet) er forskellige, mens værdierne på den afhængige variabel er ens, da vil korrelationen være svagere, end hvis dette ikke var tilfældet. Derimod kan det diskuteres, hvorvidt ties på den uafhængige skal trække ned i sammenhængens styrke. Hvis vi nemlig ikke er interesserede i selve *forklaringskraften*, dvs. i hvor meget variation i den afhængige, der skyldes (effekt fra) den uafhængige, men alene i, om den afhængige variabel systematisk ændrer værdi, hvis og kun hvis den uafhængige variabel gør det, så vil det være fornuftigt at se bort fra ties på den uafhængige variabel (dvs. ikke lade disse 'ties' trække ned i sammenhængsstyrken). Formlen for en asymmetrisk Somers' d er som følger:

$$Somers'\, d_y = \frac{P - Q}{P + Q + T_Y}$$

hvor T_Y er lig antal ties på den afhængige variabel, hvor der ikke samtidig er ties på den uafhængige.[28] Værdien angiver andelen af overvægt af konkordante par over diskordante, blandt par der ikke har tie på den uafhængige variabel.

Selvom der altså kan argumenteres for at inddrage ties, i hvert fald på den afhængige variabel, så er Gamma-koefficienten givet vis den mest benyttede i samfundsvidenskabelig forskning af de nævnte korrelationskoefficienter. I mange surveyundersøgelser vil variablerne da også kun kunne antage et meget begrænset antal værdier, og der vil derfor uundgåeligt forekomme en del ties. Men om dette lige frem kan bruges som argument for at benytte Gamma, er tvivlsomt.

Gamma, Somers' d samt en række andre koefficienter kan beregnes og udskrives i forbindelse med krydstabeller - dvs. i *proc freq*. I programeksempel 6.3 vises, hvordan dette gøres.

```
* Programeksempel 6.3;
proc freq data=SKOLE.ELEVER2;
   tables       NEWAGE NEWHGHT NEWWGHT
                NEWAGE*(NEWHGHT NEWWGHT) NEWHGHT*NEWWGHT / measures;
run;
```

Programeksemplet er identisk med eksempel 20, bortset fra at jeg her ikke beder om Chi-square baserede test.

[28] Hvis man mener, at Gamma-koefficienten giver et forkert billede af sammenhængen, og man samtidig ikke har nogen klar fornemmelse af, hvilken vej kausaliteten vender, eller hvis man er interesseret i et mål for forklaringskraft, så kan den *symmetriske* korrelationskoefficient *Kendall's tau b* evt. benyttes i stedet for Somers' d (Somers' d findes dog også i en symmetrisk version, hvor der i brøkens nævner benyttes et gennemsnit af værdierne for de to asymmetriske mål, men denne udregnes ikke af SAS). Kendall's tau b indregner ties på både den afhængige og uafhængige variabel med formlen:

$$\tau_b = \frac{P - Q}{\sqrt{(P + Q + T_X)(P + Q + T_Y)}}$$

Kendall's tau b siger, som nævnt, noget om forklaringskraften, mens en asymmetrisk Somers' d siger noget om variationen i den afhængige variabel, når og kun når den uafhængige variabel ændrer værdi.

STATISTICS FOR TABLE OF NEWHGHT BY NEWWGHT

Statistic	Value	ASE
Gamma	0.890	0.019
Kendall's Tau-b	0.669	0.024
Stuart's Tau-c	0.662	0.026
Somers' D C|R	0.663	0.026
Somers' D R|C	0.674	0.023
Pearson Correlation	0.726	0.023
Spearman Correlation	0.727	0.023
Lambda Asymmetric C|R	0.389	0.059
Lambda Asymmetric R|C	0.446	0.045
Lambda Symmetric	0.419	0.048
Uncertainty Coefficient C|R	0.329	0.023
Uncertainty Coefficient R|C	0.325	0.024
Uncertainty Coefficient Symmetric	0.327	0.024

Effective Sample Size = 366
Frequency Missing = 1

Note: Når der f.eks. skrives '`Somers' D C|R`', så betyder det, at kolonnevariablen er den afhængige variabel og rækkevariablen den uafhængige.

Gamma-koefficientens *value* er på 0,89, hvilket signallerer stærk positiv sammenhæng, og *ASE* (asymptotisk standardfejl) er på 0,019. SAS udskriver ikke sikkerheden for, at *Gamma*-værdien er forskellig fra 0, men denne er ganske nem at beregne ved hjælp af standardfejlen. Ved at dividere standardfejlen op i værdien (0,89/0,019) fås en z-værdi (antal standardvariationer), og dette tal kan slås op i tabellen over standardnormalfordelingen - bagerst i de fleste statistikbøger. I dette tilfælde kommer vi op på en z-værdi på ca. 47, hvilket i praksis betyder, at Gamma-værdien *helt sikkert* er større en '0'. Vær opmærksom på, at testen er énsidig. Hvis der ikke på forhånd er nogen antagelse om sammenhængens *retning*, må den fundne signifikans fordobles for at få den tosidede sikkerhed.

Somers' d har som ventet en noget lavere værdi (pga. de indregnede ties i den afhængige variabel). Her kan ovennævnte beregninger for signifikans-niveau naturligvis bruges på samme vis, og formlen ser ud som følger:

$$Z = \frac{\text{estimat} - \text{nulhypotese}}{\text{standardfejl}}$$

I eksemplet ovenfor er nulhypotesen lig med '0', da jeg tester for, om estimatet - altså værdien af korrelationskoefficienten - er større end '0'. Derfor kunne nulhypotesen i praksis udelades af formlen, men det ses altså, at man kan teste, hvorvidt estimatet er signifikant større end enhver given størrelse.

Vil man i stedet sammenligne to estimater, og teste om disse er signifikant forskellige, kan man benytte følgende formel:

$$Z = \frac{(\text{estimat1} - \text{estimat2})}{\sqrt{\text{standardfejl1}^2 - \text{standardfejl2}^2}}$$

Man tester altså her, om *forskellen* er signifikant forskellig fra '0'. Dette kan være en relevant beregning, hvis man f.eks. har gentaget en tidligere undersøgelse og vil teste for ændringer, eller hvis man vil teste forskellen mellem forskellige delpopulationer. Der kan i øvrigt også beregnes et 95 pct. sikkerhedsinterval for estimatet ved at multiplicere standardfejlen med 1,96 og henholdsvis trække det fundne tal fra og lægge det til estimatets værdi:

$$95\,\text{pct. sikkerhedsinterval} = \text{estimat} \pm (1{,}96 \times \text{standardfejl})$$

***Hvordan man gør i ASSIST*:**
Se ASSIST-vejledningen under afsnit 6.1.

Lineære sammenhænge mellem to variabler:
Korrelationskoefficienten *Pearson r* måler den *lineære* sammenhæng mellem to variabler, og denne skal kort omtales. Her bruges ikke rangordenen, men derimod variablenes covarians samt deres standardvariationer. Der kræves derfor her, at begge variabler er interval- eller ratioskaleret og formelt også normalitet i fordeling, og formlen ser således ud[29]:

$$Pearson\,r = \frac{[(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{(x_i - \bar{x})^2 \quad (y_i - \bar{y})^2}}$$

[29] Som det fremgår af formlen, divideres kovariansen med produktet af de to standardvariationer - alt i alt *den standardiserede kovarians*. Derfor vil den også kunne skrives som: $Pearson\,r = \frac{S_{xy}}{S_x S_y}$

hvor x_i og y_i er værdien af henholdsvis *x* og *y* i den i'te observation, og $\bar{x}$ og $\bar{y}$ er den gennemsnitlige værdi af henholdsvis *x* og *y*. Pearson r er den standardiserede kovarians mellem de to variabler *x* og *y*. Vær bl.a. opmærksom på, at enligt liggende observationer langt væk fra gennemsnittet (outliere) øver stor indflydelse på koefficientens størrelse. En *Spearman* korrelations-koefficient kan bruges i stedet for Pearson, hvis der ønskes mindre sensibilitet over for outliere og assymetrisk fordeling. Her erstattes variablenes værdier med deres rangorden, og derefter beregnes Pearson r med disse størrelser.

Som det ses fra det ovenfor viste eksempel, så beregnes og udskrives Pearson r og Spearman i forbindelse med krydstabeller. Normalt vil man imidlertid benytte en anden procedure i denne forbindelse, i og med det som oftest vil være uinteressant - og mange gange direkte forvirrende - at få udskrevet krydstabeller over sammenhænge mellem interval- eller ratioskalerede variabler. Disse kan jo - i hvert fald i princippet - antage et uendelig stort antal værdier. Man bruger i stedet proc corr. En stor fordel i denne forbindelse er i øvrigt, at man i proc corr kan få printet såkaldte korrelations-matricer ud, hvor de bivariate sammenhænge mellem en række variabler vises. Som før nævnt, kan rangkorrelationskoefficienten Gamma desværre ikke printes ud på denne vis; det kan derimod Kendall's tau b, som ligeledes benyttes til ordinalskalerede variable. Herunder viser vi, hvordan de bivariate Pearson korrelationskoefficienter for variablerne 'AGE', 'HEIGHT' og 'WEIGHT' beregnes og udskrives. Jeg benytter mig selvfølgelig her af de oprindelige ratioskalerede variabler og ikke de rekodede, ordinalskalerede som i eksemplet ovenfor.

```
* Programeksempel 6.4;
proc corr data=skole.elever2
   pearson;
   var AGE HEIGHT WEIGHT;
run;
```

Og programmet giver følgende udskrift:

```
Simple Statistics

Variabler             N          Mean       Std Dev           Sum       Minimum       Maximum

AGE                 367     13.057221      2.141062   4792.000000     10.000000     16.000000
HEIGHT              367    161.435967     14.744661         59247    132.000000    191.000000
WEIGHT              366     58.049180     11.930979         21246     31.000000     90.000000

  Pearson Correlation Coefficients / Prob > |R| under Ho: Rho=0 / Number of Observations

                                   AGE            HEIGHT            WEIGHT

               AGE              1.00000           0.56904           0.50285
                                 0.0               0.0001            0.0001
                                   367               367               366

               HEIGHT           0.56904           1.00000           0.85538
                                 0.0001            0.0               0.0001
                                   367               367               366

               WEIGHT           0.50285           0.85538           1.00000
                                 0.0001            0.0001            0.0
                                   366               366               366
```

Øverst i udprintet findes nogle univariate statistikker, og nederst findes selve korrelationsmatricen. Som overskrift til matricen er det noteret, hvad de enkelte tal står for - foruden selve Pearson r koefficienten er det altså det to-sidede signifikansniveau (sandsynlighed for at korrelationen er lig '0' i populationen, givet den absolutte værdi af den beregnede koefficient) samt antallet af observationer i analysen.

Der går en linie diagonalt gennem matricen fra øverste venstre hjørne til nederste højre hjørne, hvor korrelationskoefficienterne er lig med '+1' - altså perfekt positiv sammenhæng, hvilket er indlysende, da de tre variabler her bliver korreleret med sig selv. På hver side af denne linie afspejles den modstående side, og der er altså reelt kun tre interessante koefficientstørrelser - en for hver af de tre mulige bivariate sammenhænge. I den nederste linie midt for ses det, at Pearson r for sammenhængen mellem 'HEIGHT' og 'WEIGHT' er på 0,86. I det tidligere viste eksempel, hvor vi fik en Pearson r ud for sammenhængen mellem de to rekodede variabler, sås en koefficient på 0,73 - altså lidt forskellig fra den metode, hvor vi udnytter alle informationerne. På den anden side er de trods alt ret sammenlignelige i styrke, og faktisk vil der som oftest vise sig en sådan nogenlunde overensstemmelse. Mange er af den holdning, at der heraf kan drages den konsekvens, at det i en del situationer kan forsvares at benytte ordinalskalerede variabler i statistiske procedurer, der formelt kræver interval- eller ratioskalerede variabler.

I det viste eksempel printer jeg alle de mulige bivariate korrelationskoefficienter mellem de listede variabler. I mange tilfælde er man kun interesseret i nogle af disse kombinationer. F.eks. vil man ofte blot have korreleret en enkelt variabel med en række andre. Hvis vi f.eks. er interesseret i at korrelere 'AGE' med 'HEIGHT' og 'WEIGHT', men ikke 'HEIGHT' med 'WEIGHT', så kan vi skrive følgende program i stedet:

```
* Programeksempel 6.5;
proc corr data=skole.elever2
   pearson;
   var AGE;
   with HEIGHT WEIGHT;
run;
```

Jeg vender tilbage til proc corr i det følgende kapitel i forbindelse med partial korrelation, hvor der ses på sammenhængen, kontrolleret for tredievariabel (dvs. korrelationen mellem to variabler, justeret for effekten fra en tredje variabel – dette kaldes også for den *specificerede* sammenhæng), men først i dette kapitel skal jeg behandle den mere grundlæggende trivariate analyse ved hjælp af krydstabeller.

***Hvordan man gør i ASSIST (bivariat sammenhæng, korrelation)*:**
Vælg 'Primary menu' - 'DATA ANALYSIS' - 'ELEMENTARY' - 'Correlation'. Vælg herefter 'Active data set' og 'Variables to be correlated'. Derpå klikkes der på de variabler, der skal med i korrelationsmatricen. Sørg også for, at de/den korrekte korrelationskoefficient(er) er afkrydset, og hvis der skal foretages partial korrelation, gøres dette ved at klikke på 'Additional options' og dernæst 'Partial correlation variables'.

7.TRIVARIAT SAMMENHÆNG

Vi skal nu se på, hvad der sker med en sammenhæng mellem to variabler, når der inddrages en tredjevariabel. Først ser vi i afsnit 7.1. på krydstabeller med tilhørende korrelations-koefficienter, hvor der kontrolleres for tredjevariabel, hvorefter vi i afsnit 7.2. ser på partielle korrelationskoefficienter. De teknikker, der beskrives her må betragtes som indledende. I de fleste tilfælde vil man med fordel kunne anvende enten regressionsanalyse med dummy-variable eller loglineære analyser, men disse analyse-former kræver mere statistisk viden end forudsat blandt læserne af dette notat[30].

7.1. Krydstabeller med kontrol for tredjevariabel.

I sidste afsnit så vi, hvordan der var en signifikant og stærk sammenhæng mellem højde og vægt, hvilket for så vidt også var meget forventeligt. Jeg vil nu undersøge en ny primær sammenhæng mellem to variabler, nemlig mellem alder og vægt. Vi ved jo næsten uden at undersøge det nøjere, at der findes en bivariat sammenhæng mellem disse to variabler, fordi vores stikprøve består af skoleelever i aldersklasser, hvor der sker en stor vækst (mellem 11 og 16 år), og vi har jo lige set, at jo højere eleverne er, des tungere er de også. Men lad os sige, at vi har en teori om, at jo ældre eleven er, desto mere er der tendens til overvægt i forhold til højden. Problemet er altså at finde ud af, hvordan vi skiller os af med den helt naturlige og logiske sammenhæng, der findes mellem alder og vægt, sådan at vi står tilbage med den rest-effekt, der ikke angår højden.

Dette kan ganske simpelt gøres ved at se på den bivariate sammenhæng mellem alder og vægt, når vi holder højden fast på samme niveau. Inden for de enkelte kategorier af højde (kontrolvariablen) ser vi på den oprindelige sammenhæng mellem alder og vægt. Hvis ikke der er nogen resteffekt, så vil der ingen sammenhæng kunne ses i disse enkelte delsammenhænge. En ren *bivariat* analyse af sammenhængen mellem alder og vægt vil altså ikke kunne sige noget om teorien vedrørende stigende overvægt med stigende alder. Jeg prøver dog alligevel først at se, hvordan denne bivariate sammenhæng ser ud:

[30] Relevante procedurer i SAS er bl.a.: proc glm, proc catmod og proc factor.

```
*Programeksempel 7.1;
proc freq data=skole.elever2;
   tables newage*newwght / chisq measures ;
run;
```

Krydstabellen og en del af de statistiske mål vises herunder:

```
                TABLE OF NEWAGE BY NEWWGHT

NEWAGE(Alder - tre kategorier)
                  NEWWGHT(Vægt - tre kategorier)
Frequency,
Percent  ,
Row Pct  ,
Col Pct  ,Let     ,Middel  ,Tung    ,  Total
ffffffffˆffffffffˆffffffffˆffffffffˆ
Ung      ,     61 ,     51 ,     10 ,    122
         ,  16.67 ,  13.93 ,   2.73 ,  33.33
         ,  50.00 ,  41.80 ,   8.20 ,
         ,  55.96 ,  32.90 ,   9.80 ,
ffffffffˆffffffffˆffffffffˆffffffffˆ
Ældre    ,     28 ,     63 ,     31 ,    122
         ,   7.65 ,  17.21 ,   8.47 ,  33.33
         ,  22.95 ,  51.64 ,  25.41 ,
         ,  25.69 ,  40.65 ,  30.39 ,
ffffffffˆffffffffˆffffffffˆffffffffˆ
Ældst    ,     20 ,     41 ,     61 ,    122
         ,   5.46 ,  11.20 ,  16.67 ,  33.33
         ,  16.39 ,  33.61 ,  50.00 ,
         ,  18.35 ,  26.45 ,  59.80 ,
ffffffffˆffffffffˆffffffffˆffffffffˆ
Total         109      155      102      366
           29.78    42.35    27.87   100.00

Frequency Missing = 1


      STATISTICS FOR TABLE OF NEWAGE BY NEWWGHT

Statistic                              Value        ASE
fffffffffffffffffffffffffffffffffffffffffffffffffffff
Gamma                                  0.533      0.056
Kendall's Tau-b                        0.367      0.042
Stuart's Tau-c                         0.363      0.042

Somers' D C|R                          0.363      0.042
Somers' D R|C                          0.370      0.042
```

Vi ser en tydelig positiv sammenhæng, således at jo ældre desto større vægt - Gamma værdien er på 0,53 og Somers' d på 0,36 (en middelstærk sammenhæng). For at checke vores teori, kontrollerer jeg nu denne sammenhæng for højde. Dette gøres ved at indsætte kontrolvariablen i *proc freq* umiddelbart før den uafhængige variabel på følgende måde[31]:

```
* Programeksempel 7.2;
proc freq data=skole.elever2;
   tables newhght*newage*newwght / chisq measures;
run;
```

[31] Det er naturligvis muligt at inddrage flere kontrolvariabler ved at forlænge rækken af variabler med stjernetegn mellem - bare man husker, at variablerne i den primære sammenhæng, som man vil elaborere, kommer til sidst i rækken. Det bliver dog meget hurtigt meget uoverskueligt, og desuden vil man som regel også meget hurtigt have for få forventede værdier i de enkelte celler i krydstabellerne. Krydstabeller egner sig derfor ikke til samtidig kontrol for mange variabler.

Der viser sig imidlertid et problem her, for i den første og sidste af de tre tabeller er en af kolonnerne helt tomme, og SAS regner derfor ingen statistiske mål ud for disse tabeller. Men alene ud fra udskriften (er ikke vist) af de tre tabeller er der intet, der tyder på, at alderen skulle have nogen direkte effekt på vægten, som ikke samtidig angår højden. For at vise tabellerne her, uden at de fylder flere sider, kører jeg følgende program, hvor jeg undertrykker udskrift af frekvenser, procenter og kolonneprocenter, og jeg får derfor alene skrevet rækkeprocenterne ud, som vist umiddelbart efter programmet:

```
*Programeksempel 7.3;
proc freq data=skole.elever2;
   tables newhght*newage*newwght / nofreq nopercent nocol chisq measures;
run;
```

```
        TABLE 1 OF NEWAGE BY NEWWGHT
        CONTROLLING FOR NEWHGHT=Lav

NEWAGE(Alder - tre kategorier)
         NEWWGHT(Vægt - tre kategorier)
Row Pct ,Let     ,Middel  ,Tung    ,  Total
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Ung     ,  69.74 ,  30.26 ,   0.00 ,
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Ældre   ,  54.84 ,  45.16 ,   0.00 ,
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Ældst   ,  83.33 ,  16.67 ,   0.00 ,
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Total         80       39        0      119

Frequency Missing = 1


        TABLE 2 OF NEWAGE BY NEWWGHT
       CONTROLLING FOR NEWHGHT=Middel

NEWAGE(Alder - tre kategorier)
         NEWWGHT(Vægt - tre kategorier)
Row Pct ,Let     ,Middel  ,Tung    ,  Total
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Ung     ,  26.67 ,  66.67 ,   6.67 ,
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Ældre   ,  23.40 ,  68.09 ,   8.51 ,
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Ældst   ,  27.03 ,  48.65 ,  24.32 ,
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Total         29       70       15      114


        TABLE 3 OF NEWAGE BY NEWWGHT
        CONTROLLING FOR NEWHGHT=Høj

NEWAGE(Alder - tre kategorier)
         NEWWGHT(Vægt - tre kategorier)
Row Pct ,Let     ,Middel  ,Tung    ,  Total
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Ung     ,   0.00 ,  50.00 ,  50.00 ,
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Ældre   ,   0.00 ,  38.64 ,  61.36 ,
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Ældst   ,   0.00 ,  28.77 ,  71.23 ,
ƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Total          0       46       87      133
```

I de sidste to tabeller *kunne* der måske være tale om en positiv sammenhæng. Havde jeg imidlertid vist de fulde tabeller, ville det fremgå, at den tilsyneladende sammenhæng kunne skyldes tilfældigheder, fordi det *nominelle* antal i mange af cellerne er forholdsvis lille. Jeg er derfor nødt til at have nogle statistiske mål for sammenhængens sikkerhed og styrke. For at beregne chisquare test og Gamma er jeg nødt til at smide de to nævnte kolonner ud. I programeksempel 7.4 viser jeg, hvordan det gøres. Og samtidig med udtrækkelsen af de to kolonner beder jeg SAS om ikke at printe tabellerne ud, da det nu kun er de statistiske mål, jeg er interesseret i. Jeg er nødt til at lave tre proc freq (én for hver værdi på kontrolvariablen), da det dels kun er i to af tabellerne, der skal pilles en kolonne ud, dels fordi det er forskellige kolonner, der skal pilles ud - i den første tabel må værdien for vægt ikke være lig med 3, og i den tredje tabel må den ikke være lig med 1.

```
*Programeksempel 7.4;
proc freq data=skole.elever2;
   where newhght eq 1 and newwght ne 3;
   tables newage*newwght / chisq measures;
run;

proc freq data=skole.elever2;
   where newhght eq 2;
   tables newage*newwght / chisq measures;
run;

proc freq data=skole.elever2;
   where newhght eq 3 and newwght ne 1;
   tables newage*newwght / chisq measures;
run;
```

Jeg har snydt lidt med output'et og viser kun Gamma- og Somers' d-værdierne med tilhørende standardfejl til de tre tabeller:

Gamma	Value	ASE
NEWAGE by NEWWGHT (NEWHGHT=1)	0.074	0.177
NEWAGE by NEWWGHT (NEWHGHT=2)	0.152	0.147
NEWAGE by NEWWGHT (NEWHGHT=3)	0.270	0.150

Somers' D C|R	Value	ASE
NEWAGE by NEWWGHT (NEWHGHT=1)	0.033	0.081
NEWAGE by NEWWGHT (NEWHGHT=2)	0.083	0.082
NEWAGE by NEWWGHT (NEWHGHT=3)	0.127	0.074

Disse koefficienter skal sammenlignes med de oprindelige, som jeg gentager herunder:

STATISTICS FOR TABLE OF NEWAGE BY NEWWGHT

Statistic	Value	ASE
Gamma	0.533	0.056
Somers' D C|R	0.363	0.042

Det er tydeligt også fra sammenligningen mellem de kontrollerede og de bivariate korrelations-koefficienter, at langt overvejende er sammenhængen mellem alder og vægt forsvundet efter kontrollen for højde[32]. Kun i den sidste kategori for højde kunne der med en rimelig sikkerhed se ud til at være en moderat sammenhæng. Gamma er her 0,27 med en Z-værdi på 1,8 (0,27/0,15). Efter opslag i tabel for standard normalfordelingen findes et signifikansniveau på 0,036. Problemet er bare, at en sammenhæng alene i denne del-population ikke giver nogen mening i forbindelse med vores teori. Det virker ikke på nogen måde logisk, at hypotesen om stigende overvægtsproblemer med stigende alder *alene* skulle gælde for høje elever. Og når samtidigt at signifikansniveauet ikke er overvældende godt, vælger jeg at forkaste hypotesen.

Eksemplet ovenfor viser, at man ikke uden videre skal acceptere enhver funden sammenhæng, blot signifikansniveauet er 0,5 eller derunder. Om man kan stole på en sammenhæng afhænger foruden den statistiske sikkerhed tillige af, om sammen-hængen virker logisk, om der er teoretisk belæg for den, samt om den kan støttes af tidligere empiri (og/eller andre sammenhænge i samme undersøgelse). Hvis slet ikke den kan understøttes af nogen af disse ting, kan det ende med, at man må forkaste den og tilskrive den tilfældighedernes spil, som det er tilfældet her. Og af signifikansniveauet på 0,036 fremgår det da også, at det langt fra var helt usandsynligt, at der netop ingen sammenhæng er i populationen af høje elever. Fundet *kunne* dog give anledning til nøjere undersøgelser!

Ekskurs: 'Warning' fra SAS om for stor en andel celler i krydstabel med forventet antal observationer på under fem.

Jeg har i ovenstående vist, hvordan vi kan afhjælpe problemer med 'warnings' i forbindelse med krydstabeller ved at analysere på såkaldte subset. Dette er specielt i tilfælde af at der mangler (eller næsten mangler) hele rækker og/eller kollonner. I andre tilfælde, hvor SAS advarer om for stor andel af celler i tabellen med et forventet antal observationer på under fem, men hvor det ikke er oplagt ligefrem at smide kollonner eller rækker ud, og hvor man heller ikke med fornuft vil kunne slå kategorier sammen, da kan

[32] En såkaldt *partial Gamma* (kontrolleret korrelation) kan eksempelvis udregnes som en vægtet sum af de enkelte Gamma'er. Vægtene er ikke andel observationer i de enkelte grupper, men beregnes derimod for hver enkelt Gamma som den reciprokke varians af pågældende Gamma (dvs. 1 divideret med kvardratet af standardfejlen) divideret med summen af disse reciprokke varianser. Vægtene multipliceres nu med de enkelte Gamma-værdier, og de tre fundne størrelser summeres for at udregne den partielle Gamma. I ovennævnte eksempel findes en partial Gamma på 0,17. Det kan endvidere testes, hvorvidt den partielle Gamma er signifikant forskellig fra nul. Variansen til den partielle Gamma findes som summen af: de kvardrerede vægte multipliceret med de tilhørende enkelte varianser. Herefter findes en Z-værdi som den partielle Gamma divideret med standardfejlen til denne (kvardratroden af variansen til den partielle Gamma). Signifikansen findes nu i normalfordelingstabellen. (Se i øvrigt mere herom i Kreiner (1999) p. 337-39.)

problemernes omfang i stedet vurderes nøjere. Dette kan f.eks. gøres ved at undersøge, hvor stort et bidrag de kritiske tabel-celler med forventet antal under fem giver til Chi-square Value.

I det trivariate eksempel ovenfor kunne denne fremgangsmåde være benyttet i forbindelse med den midterste tabel. I tilknytning til denne tabel bliver der nemlig angivet, at 22 pct. af cellerne har en forventet værdi på under fem. Nedenstående program viser, hvordan vi kan få printet tal ud for de enkelte cellers bidrag til Chi-square Value ('cellchi2'). Desuden bedes i programmet om tal for forventet antal ('expected'), og jeg undertrykker tallene for procenterne samt kollonneprocenterne ('nopercent' og 'nocol') for ikke at få en alt for uoverskuelig tabel. (I *tables*-sætningen kunne jeg nøjes med at skrive 'NEWAGE*NEWWGHT', da jeg ovenfor har bedt om kun at få tabellen ud for observationer, hvor 'NEWHGHT' er lig med '2', men for at få en sigende overskrift ud, vælger jeg alligevel at sætte 'NEWHGHT' ind som kontrolvariabel.)

```
*Programeksempel 7.5;
proc freq data=LOLLE.ELEVER2  ;
   where NEWHGHT=2;
   tables NEWHGHT*NEWAGE*NEWWGHT / nopercent nocol expected cellchi2 chisq;
run;
```

Vi får efter kørsel af programmet følgende tabeludskrift og statistiske mål:

```
          TABLE 1 OF NEWAGE BY NEWWGHT
         CONTROLLING FOR NEWHGHT=Middel

NEWAGE(Alder - tre kategorier)     NEWWGHT(Vægt - tre kategorier)

Frequency      ,
Expected       ,
Cell Chi-Square,
Row Pct        ,Let     ,Middel  ,Tung    ,  Total
ƒƒƒƒƒƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆ
Ung            ,       8 ,     20 ,      2 ,     30
               , 7.6316 , 18.421 , 3.9474 ,
               , 0.0178 , 0.1353 , 0.9607 ,
               ,  26.67 ,  66.67 ,   6.67 ,
ƒƒƒƒƒƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆ
Ældre          ,      11 ,     32 ,      4 ,     47
               , 11.956 ,  28.86 , 6.1842 ,
               , 0.0765 , 0.3417 , 0.7714 ,
               ,  23.40 ,  68.09 ,   8.51 ,
ƒƒƒƒƒƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆ
Ældst          ,      10 ,     18 ,      9 ,     37
               , 9.4123 , 22.719 , 4.8684 ,
               , 0.0367 , 0.9803 , 3.5063 ,
               ,  27.03 ,  48.65 ,  24.32 ,
ƒƒƒƒƒƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒˆ
Total               29       70       15      114
```

```
     STATISTICS FOR TABLE 1 OF NEWAGE BY NEWWGHT
           CONTROLLING FOR NEWHGHT=Middel

Statistic                    DF     Value        Prob
ffffffffffffffffffffffffffffffffffffffffffffffffffffffff
Chi-Square                    4     6.827       0.145
Likelihood Ratio Chi-Square   4     6.505       0.164
Mantel-Haenszel Chi-Square    1     1.378       0.240
Phi Coefficient                     0.245
Contingency Coefficient             0.238
Cramer's V                          0.173

Sample Size = 114
WARNING: 22% of the cells have expected counts less
         than 5. Chi-Square may not be a valid test.
```

For det første ses det, at Chi-square testen ikke har et tilstrækkelig lavt signifikansniveau til at vi kan forkaste en hypotese om ingen sammenhæng. Det kan diskuteres, hvorvidt det er en god ide eller ej at blive ved med at slå kategorier sammen i sin søgen efter acceptable signifikansniveauer, men *hvis* vi slår kategorierne 'Ung' og '*Æ*ldre' samt 'Let' og 'Middel' sammen, hvorved fås en firefelts-tabel, finder vi et fint signifikansniveau på 0,014. Stadigvæk er der dog en 'warning' om, at over 20 pct. af cellerne har en forventet værdi på under fem, så dét problem bliver altså ikke løst ved rekodningen, og der kan umuligt slås flere kategorier sammen. Tabellen vises herunder (programmerne til rekodning og udskrift vises ikke, da de følger samme principper, som tidligere viste eksempler.

```
      TABLE 1 OF NEWAGE BY NEWWGHT
     CONTROLLING FOR NEWHGHT=Middel

NEWAGE(Alder - tre kategorier)
              NEWWGHT(Vægt - tre kategorier)
Frequency     ,
Expected      ,
Cell Chi-Square,
Row Pct       ,Let     ,Tung    ,  Total
ffffffffffffffˆffffffffˆffffffffˆ
Yngre         ,     71 ,      6 ,     77
              , 66.868 , 10.132 ,
              , 0.2553 , 1.6848 ,
              ,  92.21 ,   7.79 ,
ffffffffffffffˆffffffffˆffffffffˆ
Gammel        ,     28 ,      9 ,     37
              , 32.132 , 4.8684 ,
              , 0.5313 , 3.5063 ,
              ,  75.68 ,  24.32 ,
ffffffffffffffˆffffffffˆffffffffˆ
Total              99       15      114
```

```
     STATISTICS FOR TABLE 1 OF NEWAGE BY NEWWGHT
           CONTROLLING FOR NEWHGHT=Middel

Statistic                      DF     Value        Prob
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
Chi-Square                      1     5.978       0.014
Likelihood Ratio Chi-Square     1     5.579       0.018
Continuity Adj. Chi-Square      1     4.618       0.032
Mantel-Haenszel Chi-Square      1     5.925       0.015
Fisher's Exact Test (Left)                        0.996
                    (Right)                       0.018
                    (2-Tail)                      0.020
Phi Coefficient                       0.229
Contingency Coefficient               0.223
Cramer's V                            0.229

Sample Size = 114
WARNING: 25% of the cells have expected counts less
         than 5. Chi-Square may not be a valid test.
```

Den enlige celle med et forventet antal observationer på under fem har en Chi-square værdi (‘Cell Chi-Square’) på 3,5063, hvilket er mere, end hvad cellerne med fem og derover i forventet antal udgør, nemlig 2,4717. Dette er ikke nogen ideel situation, og det vil være forbundet med nogen usikkerhed at forkaste nul-hypotesen om uafhængighed. Dog skal det nævnes i forbindelse med advarslen mod at benytte Chi Square testen, at såvel grænsen på 20 pct. af cellerne med kritisk lavt forventet antal observationer som fastsættelsen af fem som det laveste antal observationer, der kan betragtes som værende ikke-kritisk, kan diskuteres. Således taler nogle for at andelen af celler med kritisk lavt antal forventede observationer kan sættes højere end de 20 pct., uden at det udløser advarsel, og andre taler for at sænke grænsen fra fem til tre for et acceptabelt niveau i hver celle. Begge dele ville hver for sig medføre, at vi ville acceptere Chi Square testen i ovennævnte eksempel, men under alle omstændigheder ligger eksemplet i grænselandet, hvor der ikke findes éntydige regler.

Angående kausalitet:

I forbindelse med eksemplet med kontrol for tredjevariabel skal der her til slut gøres et par vigtige tilføjelser. Før man inddrager kontrolvariabler, er det vigtigt, at man har gjort sig overvejelser om, hvilke variabler der potentielt kunne være relevante. Kun variabler, der kausalt/tidsmæssigt ligger tidligere end - eller mindst på linje med - den afhængige variabel kan komme på tale. Ligger kontrolvariablen kausalt/tidsmæssigt *efter* eller på linje med den uafhængige variabel, er der tale om *specificering* af sammenhængen, og ikke om hvorvidt sammenhængen er ægte eller spuriøs.

I det her omtalte eksempel ligger kontrolvariablen, højde, netop kausalt og tidsmæssigt efter den uafhængige, og det kan diskuteres, hvorvidt den ligger før eller på linie med den afhængige. Sammenhængen findes altså, og vi vil blot se, om der er en

direkte effekt fra alder, som kan skilles ud fra den helt naturlige effekt på såvel højde som vægt – en effekt som kan kaldes overvægts-effekt. I mange situationer vil det også være af interesse, om sammenhængen er forskellig fra tabel til tabel efter kontrollen. I eksemplet her var der faktisk forskel mellem tabellerne, men på baggrund af teoretiske overvejelser blev sammenhængen mellem alder og vægt blandt de høje forkastet. I andre situationer med andre variabler og problemstillinger vil det selvfølgelig ofte være plausibelt, at der er forskel i sammenhængen afhængigt af værdien på kontrolvariablen. Det vil være for omfattende at komme ind på alle de mange forskelligartede situationer, man vil kunne komme ud for, men der kan henvises til Kreiner (1999) kap. 12 og 13 samt til Rosenberg (1968).

***Hvordan man gør i ASSIST (trivariat sammenhæng, krydstabeller)*:**
Der gøres som under bivariat analyse, blot med én tilføjelse: Når man skal vælge analysevariabler, så skal kontrolvariablen vælges først, dernæst den uafhængige og tilsidst den afhængige. Kontrolvariablen skal altid vælges først, uanset hvordan den kausalt eller tidsmæssigt ligger i forhold til de øvrige variabler. Under 'Crosstabulations' klikkes på linjen med alle tre variabler - og evt. andre. Resten foregår som under bivariat analyse.

7.2. Partial korrelation.

Jeg vil nu ganske kort vise, hvordan man ved hjælp af *partial korrelation* nemt kan foretage samme kontrol som i ovenstående krydstabel-eksempel. Partial korrelation kan på ingen måde erstatte analyse med krydstabeller, i og med at der hermed udelades en stor mængde detaljeret information, som man ofte vil være interesseret i. I mange tilfælde er man imidlertid kun interesseret i en komprimeret information om sammenhængene, og derforuden har vi set, at kontrollerede sammenhænge i krydstabeller meget hurtigt bliver yderst kompliceret at overskue, ligesom det kan være vanskeligt at skille systematik fra tilfældighed pga. få observationer i de enkelte tabel-celler. Især hvis variablerne i analysen har mange kategorier, bliver problemerne ofte uoverskuelige. En anden måde at sige det på, er at analysen drukner i detaljer.

Jeg viser herunder, hvordan vi kan få printet dels bivariate korrelationskoefficienter ud, dels de partielle, hvor der er kontrolleret for tredievariabel. Jeg vælger her at bruge de rekodede variabler, og jeg vælger koefficienterne Pearson r og Kendall tau b (Gamma og Somers' d kan ikke udskrives med denne procedure). Ved de partielle koefficienter bliver der kun beregnes signifikansniveau for Pearson r.

```
*Programeksempel 7.6;

*Simpel korrelation - alder og vægt - rekodede variable;
proc corr data=lolle.elever2
   pearson kendall nosimple;
   var newage;
   with newwght;
run;

*Partial korrelation - alder og vægt, kontrolleret for højde - rek. variable;
proc corr data=lolle.elever2
   pearson kendall nosimple;
   var newage;
   with newwght;
   partial newhght;
run;
```

Læg mærke til, at der efter specificeringen af korrelations-koefficienterne Pearson r og Kendall tau b skrives *nosimple*. Hermed angives, at vi ikke er interesseret i at få udskrevet de simple, univariate statistikker, men kun korrelationskoefficienterne. Programmet giver følgende udskrift (først de bivariate koefficienter, dernæst de partielle):

```
Correlation Analysis

                                1 'WITH' Variables:  NEWWGHT
                                1 'VAR'  Variables:  NEWAGE


  Pearson Correlation Coefficients / Prob > |R| under Ho: Rho=0 / Number of Observations

                                                        NEWAGE

                           NEWWGHT                     0.40559
                           Vægt - tre kategorier        0.0001
                                                           366


Kendall Tau b Correlation Coefficients / Prob > |R| under Ho: Rho=0 / Number of Observations

                                                        NEWAGE

                           NEWWGHT                     0.36685
                           Vægt - tre kategorier        0.0001
                                                           366
```

```
Correlation Analysis

                                1 'PARTIAL' Variables:  NEWHGHT
                                1 'WITH'    Variables:  NEWWGHT
                                1 'VAR'     Variables:  NEWAGE


 Pearson Partial Correlation Coefficients / Prob > |R| under Ho: Partial Rho=0 / N = 366

                                                        NEWAGE

                            NEWWGHT                    0.08464
                            Vægt - tre kategorier       0.1064


                  Kendall Partial Tau b Correlation Coefficients  / N = 366

                                                        NEWAGE

                            NEWWGHT                    0.10968
                            Vægt - tre kategorier
```

Vi ser fra disse udskrifter, at mens der er en meget sikker og også en moderat til stærk bivariat sammenhæng mellem ’NEWAGE’ og ’NEWWGHT’ (kan ses fra både Pearson r og Kendall tau b), så svinder denne ind til næsten ingenting efter kontrollen for højde. Kendall tau b er således i bivariat analyse lig med 0,37, mens den i den trivariate analyse er helt nede på 0,11. Vi ser endvidere, at Pearson r foruden at svinde meget ind bliver insignifikant efter kontrollen (og i analyse på de oprindelige variabler, som ikke vises her, er konklusionen endnu tydeligere).

***Hvordan man gør i ASSIST (trivariat sammenhæng, korrelation)*:**
Der gøres som under bivariat analyse. Blot skal der nu vælges kontrolvariabel. Dette gøres under ’additional options’, hvor der klikkes på ’Partial correlation columns’.

8.KONSTRUKTION AF INDEKS

Ofte vil man komprimere oplysningerne/dataene fra flere variabler, således at der dannes én ny variabel, der indeholder et samlet mål for disse variabler. Der findes to hovedtyper af komprimerede mål for flere variabler - *additive indeks* og *typologier*.

Typologier:
I en typologi gives variablen forskellige værdier alt efter udfaldet på en række forskellige variabler. Ud fra to dikotome variabler kan der f.eks. dannes fire kategorier på en typologi-variabel - én kategori for hvert udfald (eller med andre ord: én kategori for hver celle i en krydstabel for de to variable). De to variabler kunne eksempelvis være køn og en aldersvariabel med kategorierne 'unge' og 'ældre', og vi ville få et udfaldsrum, der bestod af henholdsvis: 'unge kvinder', 'unge mænd', 'ældre kvinder' og 'ældre mænd'. Er der flere kategorier end to i de enkelte variabler, og/eller inddrages der flere variabler i typologien, da vil antallet af udfald selvfølgelig stige; men ofte vil man så slå nogle af udfalds-rummene sammen, således at selve typologi-variablen stadigvæk antager et ret begrænset antal værdier. Typologimetoden kan bl.a. være en fordel, hvis der analyseres med krydstabeller og et forholdsvis stort antal variabler. Hvis f.eks. man benytter ovenstående køn og alders-typologi som uafhængig variabel i en krydstabel, vil dette kunne sammenlignes med en trivariat krydstabels-analyse.

Indeks:
Et indeks er, som nævnt, ligeledes et samlet mål for en række enkeltvariabler, men til forskel fra typologier, vil et indeks være konstrueret sådan, at det ligger på ordinalskala, og ofte vil man endda betragte det som en intervalskaleret variabel. Overordnet set findes der to forskellige typer af indeks, nemlig *refleksive* og *ikke-refleksive*.

Ved et refleksivt indeks forsøger man at indfange en latent (skjult) variabel. Det kunne f.eks. være holdning til indvandrere, og de enkelte items i indekset kunne så være forskellige spørgsmål angående indvandrere, som respondenterne skulle erklære sig mere eller mindre enige i. Hvis enkeltspørgsmålene hver især hænger sammen med en generel holdning til indvandrere, så vil der være en stærk sammenhæng mellem disse - høje korrelations-koefficienter. Der vil så kunne dannes et refleksivt indeks, som man kunne kalde *holdning til indvandrere*. Man siger ogsaa, at det, der måles, er éndimen-

sionelt. Ofte vil dannelsen af indekset bestå af en simpel addition af de enkelte item-værdier, men herom mere i afsnit 8.2.

Hvis indvandrerholdning derimod ikke med nogen rimelighed kan siges at være én homogen størrelse, men istedet bestå af forskellige dimensioner, som kun delvist eller måske slet ikke hænger sammen med hinanden, vil det være mere problematisk at danne ét samlet indeks, man kalder for *holdning til indvandrere*. Hvis man alligevel gør det, må man i hvert fald være ret overbevist om, at man har indfanget samtlige dimensioner i holdningen til indvandrere, og at der ikke vil være tale om et refleksivt indeks. Et eksempel fra det virkelige forskerliv kunne måske her være på sin plads her:

Jørgen Goul Andersen (1998) forsøger i bogen "Borgerne og lovene" bl.a. at indfange begrebet moral i privatsfæren ved hjælp af en række items. Blandt spørgsmålene er der fire, hvor respondenterne skal svare på, i hvor høj grad de billiger henholdsvis det at *beholde fundne penge*, det at *lyve til egen fordel*, det at *undlade at gøre opmærksom på, at man har fået for meget tilbage i en forretning* samt det at *give sit barn en lussing, hvis barnet er groft ulydigt*. Hvis man nu kun havde haft det sidstnævnte spørgsmål med i en undersøgelse, og man brugte dette som indikator for moral i privatsfæren generelt, så ville man begå en stor fejl. Det viser sig nemlig, at de respondenter, der billiger korporlig afstraffelse af sit barn, slet ikke er de samme, som dem der billiger de øvrige ting. Derimod er der stærk korrelation indbyrdes mellem de øvrige tre items, og Goul Andersen kalder den latente variabel bag disse items for *moral i mellemmenneskelige relationer*. Der er flere andre items, som ikke er nævnt i ovenstående, og i alt finder Jørgen Goul Andersen tre forskellige dimensioner (eller latente variable) i privatsfærens moral. Nu skal jeg ile med at sige, at Goul Andersen fra start af var bevidst om, at der sikkert ville udskille sig lige netop de tre fundne dimensioner og ikke kun én, men eksemplet skulle alligevel kunne tjene som advarsel mod at tolke for vidt på enkeltitems.[33]

Det skulle gerne kunne ses nu, at der kan være en stor fordel i at teste for indbyrdes sammenhæng blandt items, som vi mener er udtryk for en latent variabel. Men hvis man så først har fundet en indbyrdes sammenhæng, hvorfor kan man så ikke blot benytte et enkelt af disse som udtryk for den latente variabel? Dette er selvfølgelig også bedre, end at benytte det enkelte item *uden* først at have testet for sammenhæng med de resterende items. Men da enkeltitems som oftest så at sige vil bestå af både en *generel* og en *specifik* del, så er det oplagt, at vi bedst indfanger det generelle ved at lave et samlet mål af alle variablerne. Alle de specifikke dele får herved i det samlede indeks kun en

[33] Endvidere analyserer Goul Andersen sig frem til de tre dimensioner ved hjælp af faktor-analyse (en rafineret form for korrelations-analyse), som vi ikke i denne vejledning vil komme nærmere ind på.

lille vægt. Dette kan billedligt sammenlignes med, at vi f.eks. har tre poser lakrids-konfekt, hvor der er to slags konfekt i hver pose. Der er én slags konfekt, som går igen i alle tre poser, og som udgør en betragtelig del (f.eks. mellem halvdelen og tre fjerdedele) af den samlede mængde i de enkelte poser. Den anden del af poserne er forskel fra pose til pose. Man kunne sige, at der er en generel del og en specifik del i hver pose. Hvis vi nu blot tager en enkelt pose frem og kigger på, så vil vi se store andele af både det generelle og det specifikke. Hvis vi derimod blander det hele godt sammen, så vil vi først og fremmest lægge mærke til det generelle. Der vil ganske vist være en masse andet konfekt, men det vil være noget blandet ubestemmeligt, og jo flere poser, desto mere ubestemmeligt og vagt vil det specifikke tage sig ud.

Eksemplet med lakridsposekonfekten viser, at jo flere poser (items) og jo større andel generelt konfekt (latent begreb), desto bedre bliver fremvisningen af det generelle. Eksemplet viser i øvrigt også, at det er bedst, hvis den generelle del er nogenlunde ligeligt repræsenteret i samtlige poser. Hvis f.eks. én ud af fem poser kun har en forholdsvis lille generel del, vil denne pose så at sige kontaminere sammenlægningen, og det vil være bedre at fjerne denne pose.

Ofte vil det kunne forsvares at danne ikke-refleksive indeks, blot man er opmærksom på, at der så netop ikke er en indre sammenhængskraft mellem de enkelte items. Et eksempel på et sådant, der gennem empiriske studier er fundet brugbart, er Galtungs indeks for social rang. Heri indgår bl.a. variabler for køn og alder, hvorimellem der naturligvis ingen sammenhæng er. Det, der alligevel gør, at man kan kalde målet for et indeks, er, at det er konstrueret til at ligge på ordinalskala, sådan at jo højere værdi på skalaen, des højere social rang indikerer det.

Jeg har her valgt at afgrænse ikke-refleksive indeks fra typologier ved spørgsmålet om skala-niveau - nominal eller ordinal/interval. Andre steder kan ses andre måder at foretage afgrænsning på og også helt andre termer for sammensatte mål i det hele taget. Det vil imidlertid kræve uforholdsmæssig plads at skulle redegøre nøjere for de forskellige termer og forskellige opfattelser af, hvordan disse skal defineres, og det vigtigste er da også, at man er i stand til at argumentere rationelt for dannelsen af et samlet, komprimeret mål, samt at dannelsen viser sig empirisk givtig.

I resten af kapitlet vil jeg koncentrere mig om refleksive indeks. I afsnit 8.1. gennemgåes rent teknisk, hvordan et indeks kan dannes i SAS. Dernæst vil jeg i afsnit 8.2. vise, hvordan en itemserie ved brug af Cronbach’s Alpha kan testes for reliabilitet, dvs. for indre homogenitet og for indikation for latent variabel.

8.1. Dannelse af additivt indeks i SAS.

En simpel summation er en ofte brugt metode til dannelse af et additivt indeks. Hvis jeg f.eks. vil danne et indeks af variablerne 'v01' til 'v05', så vil indekset komme til at gå fra en minimumsværdi på 5 (5*1) til en maksimumværdi på 25 (5*5). Den nye variabel kan så benyttes i den videre analyse, f.eks. i korrelations- eller regressions-analyse. Hvis det er hensigten at benytte indekset i forbindelse med krydstabeller, vil det være nødvendigt f.eks. at rekode de enkelte variabler til såkaldt dikotome variabler (dvs. variabler som kun kan antage to forskellige værdier). Hvis ikke der før sammenlægningen gøres noget sådant, vil der blive alt for mange værdier i indekset, og det vil være ubrugeligt til krydstabeller og Chi-square test. Alternativt kan man starte med at danne indekset og dernæst sammenlægge værdier i dette.

De nævnte variabler er tidligere blevet rekodet (se afsnit 5.4), således at værdien 6 for 'ved ikke' er sat til 3, 'hverken enig eller uenig', hvilket man skal sikre sig inden sammenlægning. Et andet problem er dog, hvad man skal stille op med *missing values*. Hvis der blot foretages summation uden at tage højde herfor, vil respondenter, som af den ene eller anden grund ikke har besvaret alle spørgsmål, generelt få en lavere indeks-værdi, end respondenter, der har besvaret alle spørgsmål. Normalt vil man sætte både 'ved ikke'-kategorien og *missing values* til enten midterværdien eller til gennemsnittet af de besvarede. Dog vil man sætte indeks til *missing value*, hvis der forekommer over et vist samlet antal 'ved ikke' og manglende besvarelser. Hvis der kun findes en meget lille andel ikke besvarede spørgsmål, kan man tillade sig den luksus at lade indekset være missing, blot et enkelt af spørgsmålene fra item-serien er ubesvaret. Jeg vælger i dette tilfælde med variablerne '`v01`' til '`v05`', at der maksimalt må være to ubesvarede og 'ved ikke' i alt. Rekodning og indeksberegning kan f.eks. se ud som i programeksempel 8.1.

```
*Programeksempel 8.1;
data x;
set SKOLE.ELEVER2;
if v01 in(1,2,3,4,5) then REK2v1=v01;
else REK2v1=.;
if v02 in(1,2,3,4,5) then REK2v2=v02;
else REK2v2=.;
if v03 in(1,2,3,4,5) then REK2v3=v03;
else REK2v3=.;
if v04 in(1,2,3,4,5) then REK2v4=v04;
else REK2v4=.;
if v05 in(1,2,3,4,5) then REK2v5=v05;
else REK2v5=.;

MANGLER=nmiss(of REK2v1-REK2v5);
if MANGLER>2 then SUMINDX=.;
else SUMINDX=sum(of REK2v1-REK2v5)+(MANGLER*3);

run;
```

I rekodningen ser vi en ny form. Hvis vi skal spørge, om en variabel er lig med én af værdierne fra en liste af værdier, kan vi skrive det i formen '`in (værdi1, værdi2, værdi3...)`'. I programeksemplet er det sådan, at hvis den oprindelige variabel er lig med 1,2,3,4 eller 5, så skal den nye rekodede variabel være lig med den oprindelige. Og hvis ikke dette er tilfældet, så skal den rekodede sættes lig med *missing value*. Man kunne også skrive f.eks.: '`if v01=1 or v01=2 or v01=3…..`', men ofte vil det være hurtigere med den viste form. Også her vil programmet i øvrigt kunne skrives ved brug af *arrays*.

I de tre sidste linjer før *run*-sætningen benyttes der funktioner og aritmetiske operatorer. Disse er allerede beskrevet i afsnit 5.6., men da dette afsnit har karakter af en ekskurs, vil jeg atter gennemgå dem herunder.

Funktionen '`nmiss`' sammentæller antal *missing values* i variabel-listen mellem paranteserne (man siger også, at funktionen returnerer denne værdi/dette antal), og den nydefinerede variabel '`MANGLER`' sættes lig med dette antal. Funktionen '`SUM`' sammentæller værdierne i variabel-listen, og den nydefinerede variabel '`SUMINDEX`' sættes lig med denne sum plus antal *missing values* gange 3 (3 for midterværdi). Se eksemplerne herunder:

Eksempler på hvordan programeksempel 22 vil virke i praksis

REK2V1	*REK2V2*	*REK2V3*	*REK2V4*	*REK2V5*	*SUMINDEX*
1	2	2	3	1	9
4	3	.	5	4	19
.	.	3	2	.	.

I programeksemplet er af aritmetiske operatorer kun brugt additionstegnet. Herunder vises symbolerne til fem aritmetiske operatorer.

Aritmetiske operatorer i SAS

Symbol	*Definition*
**	Exponential
*	Multiplikation
/	Division
+	Addition
-	Subtraktion

Eksempelvis vil '`v01**3`' betyde, at værdien af '`v01`' opløftes til tredje potens, og '`v01*v02`' at værdierne i de to nævnte variabler multipliceres. Regneoperationer foregår som vanligt i nævnte rækkefølge fra oven (plus og minus er dog ligestillede), og hvis rækkefølgen i en given operation skal være anderledes, indsættes paranteser (i programeksempel 8.1 er den sidste parentes i den aritmetiske beregning derfor ikke nødvendig, men den gør det mere overskueligt).

Angående funktioner så findes der i SAS en lang række, hvoraf der kun skal nævnes nogle ganske få herunder:

Eksempler på funktioner i SAS. Funktionsnavnet efterfølges i programmet af en parentes indeholdende enten et argument* eller en liste med variable/værdier. Ofte vil funktionen indgå i en forbindelse som følgende:

VARIABELNAVN=FUNKTIONSNAVN(ARGUMENT ELLER VARIABEL-LISTE);

Funktion	Beskrivelse (funktionen returnerer)
ABS	Absolut værdi af argument
MAX	Maksimum værdi blandt de listede
MIN	Minimum værdi blandt de listede
SIGN	Fortegn: -1 hvis mindre end 0; 0 hvis 0; 1 hvis større end 0
N	Antal valide værdier blandt de listede
NMISS	Antal *missing values* blandt de listede
SUM	Sum af de listede
CEIL	Mindste heltal større end eller lig med argument
FLOOR	Største heltal mindre end eller lig med argument
INT	Heltalsværdi af argument
ROUND	Afrundet værdi** af argument

* Et argument kan være enten et variabelnavn, en konstant eller et matematisk udtryk. Og et matematisk udtryk kan indeholdende variabelnavne, konstanter eller begge dele samt en eller flere aritmetiske operatorer.

** Hvis der f.eks. skrives 'round(121.453)' vil det returnerede tal blive '121.000' (husk at bruge punktum i stedet for komma). Man kan imidlertid afrunde på selvvalgt decimal i stedet. F.eks. vil 'round(121.453,.1)' returnere værdien '121.500' og 'round(121.453,.01) vil returnere værdien '121.450'.

Mange funktioner laver de samme regneoperationer, som man vil kunne klare med de aritmetiske operatorer. Der er imidlertid en meget afgørende forskel, nemlig i hvordan der tages hensyn til *missing values*. Hvis man benytter aritmetiske operatorer, så vil resultatvariablen blive sat til *missing*, hvis blot én af variablerne i det matematiske udtryk er *missing*. Ved brug af funktioner vil dette ikke ske, hvilket f.eks. gør det nødvendigt at benytte funktionen 'sum' i programeksempel 8.1 ovenfor. Se herunder for eksemplificering af forskellen.

***Missing values* i operationer med henholdsvis funktioner og aritmetiske operatorer.**

Programsætning	*Værdi af variablen 'indeks'*
`INDEKS=sum(of v01-v03);`	3
`INDEKS=sum(v01, v02, v03);`	3
`INDEKS=v01+v02+v03;`	.
`INDEKS=sum(v01, v04, v05);`	10
`INDEKS=v01+v04+v05;`	10

Variabelværdier:

```
v01 = 2
v02 = . (blank)
v03 = 1
v04 = 3
v05 = 5
```

Ud over funktioner og aritmetisk beregning er der i programeksempel 8.1 vist en ny måde at definere en variabel-liste på, nemlig med kun én bindestreg imellem. Med dette fortælles, at listen omfatter de nævnte to variabler plus de variabler, der har samme bogstav-navn, og som har afsluttende cifre liggende mellem de nævnte, og udfyldende hele intervallet - den går altså ikke, hvis der f.eks. ikke findes nogen variabel ved navn '`REK2V3`'. Til gengæld er der ikke krav om, at variablerne skal ligge placeret fysisk ved siden af hinanden i datamatricen. Se i øvrigt vedrørende de forskellige typer af variabel-lister i SAS i afsnit 5.5.

Vigtigt i forbindelse med dannelse af indeks (spejlvending af svar):

En meget vigtig ting i forbindelse med dannelsen af indeks, er at man skal sørge for, at de summerede variabler vender ens, således at f.eks. 1 signallerer meget enig i samtlige. Det er meget almindeligt i spørgeskemakonstruktionen at vende spørgsmålene forskelligt, bl.a. for at kunne teste validiteten og evt. for at kunne sortere ikke valide spørgeskemaer fra[34]. Hvis dette er tilfældet må man enten før eller under summationen vende nogle af spørgsmålene. Hvis vi forestiller os, at variablerne 'v02' og 'v04' vender omvendt af de

[34] Af og til stiller man således spørgsmål, som man *logisk set* vil kende svaret på, hvis man kender svaret på et givet andet spørgsmål. F.eks. burde en respondent ikke kunne være enig i et udsagn om, at indvandringen truer det danske samfund, samtidig med at samme respondent er enig i et udsagn om, at indvandringen giver et positivt input til udviklingen af det danske samfund.

øvrige i eksemplet fra før, så kunne en indeksdannelse se ud som i programeksempel 8.2 i stedet (spejlvendingen kunne for den sags skyld også foretages i sætningen med selve den aritmetiske operation, hvilket ville være en fordel, i fald vi skulle bruge de rekodede variabler til andet end indeksdannelse). I programeksemplet viser jeg samtidigt, hvordan vi kunne tildele missing values og 'Ved ikke'-besvarelser den *gennemsnitlige* værdi af de besvarede i stedet for at give dem midter-værdien '3'.

```
*Programeksempel 8.2;
data x;
set SKOLE.ELEVER2;
if V01 in(1,2,3,4,5) then REK2V1=V01;
else REK2V1=.;
if V02 in(1,2,3,4,5) then REK2V2=6-V02;
else REK2V2=.;
if V03 in(1,2,3,4,5) then REK2V3=V03;
else REK2V3=.;
if V04 in(1,2,3,4,5) then REK2V4=6-V04;
else REK2V4=.;
if V05 in(1,2,3,4,5) then REK2V5=V05;
else REK2V5=.;

MANGLER=nmiss(of REK2V1-REK2V5);
if MANGLER>2 then SUMINDX=.;
else SUMINDX=sum(of REK2V1-REK2V5)+(MANGLER*mean(of REK2V1-REK2V5));

run;
```

Læg mærke til måden, hvorpå jeg spejlvender svarene til spørgsmål to og fire. Der er fem valide svarkategorier, så hvis vi trækker værdien fra tallet '6', får vi det spejlvendte (1 bliver til 5; 2 til 4 osv.). Hvilket tal, værdien skal trækkes fra, afhænger selvfølgelig af, hvor mange værdier variablen kan antage. Var der kun fire valide værdier (1-4), så ville vi istedet trække værdien fra tallet '5'.

Jeg har nu vist den rent tekniske procedure for dannelse af et additivt indeks, men vi ved endnu ikke, om der rent faktisk er tale om et *refleksivt* indeks, hvor der er en indre sammenhængskraft. Jeg viser i afsnit 8.2. herunder, hvordan vi kan teste herfor ved hjælp af *Cronbach's Alpha*.

8.2. Reliabilitetstest - Cronbach's alpha.

Cronbach's alpha er den mest benyttede test for *reliabilitet* i itemserie til dannelse af additivt indeks. Alpha-koefficienten kan antage værdier fra '0' til '+1', og teknisk set er den et udtryk for, hvor stor sammenhængskraft der er indbyrdes mellem de enkelte items i serien. Den kan imidlertid tolkes som andel forklaret varians i den latente variabel, hvor '+1' svarer til, at den latente variabel indfanges perfekt. De fleste sætter 0,7 som et

minimumskriterium for tilstrækkelig grad af reliabilitet. Formlen for Cronbach's Alpha er:

$$\alpha = \frac{k\overline{\text{cov}} / \overline{\text{var}}}{1 + (k-1)\overline{\text{cov}} / \overline{\text{var}}}$$

hvor k er lig antal items, $\overline{\text{cov}}$ er lig med den gennemsnitlige kovarians mellem to items fra en serie, og $\overline{\text{var}}$ er lig med den gennemsnitlige varians i de enkelte item. Ud fra formlen kan det ses, at Alpha-værdien vil stige med stigning i forholdet mellem den gennemsnitlige kovarians og varians (dvs. med stigende gennemsnitlig korrelation mellem item) samt også med stigning i antallet af inddragede item.

Alpha-koefficienten beregnes og udskrives ved at indsætte en alpha-option i *proc corr*, som jeg har beskrevet i afsnit 6.2. Inden dette gøres er det dog vigtigt, at man har sikret sig, at alle items "vender" ens - dvs. at der forventes *positiv* korrelation mellem de enkelte items. I afsnit 8.2. vises, hvordan dette kan gøres. Derudover er det selvfølgelig vigtigt, at der teoretisk (eller på anden baggrund) er belæg for at opstille en hypotese om indre sammenhæng mellem variablerne i indekset. Dette problem vil jeg imidlertid ikke behandle her.

I nedenstående program vises, hvordan vi beregner alpha for variablerne 'REK2V1' til 'REK2V5', som blev dannet i afsnit 8.1., og som blev brugt til indeksdannelse:

```
*Programeksempel 8.2;
proc corr data=SKOLE.ELEVER2
   alpha kendall;
   var REK2V1-REK2V5;
run;
```

Når vi beder om alpha-koefficient, får vi automatisk udskrevet univariate statistikker samt korrelations-matrice med Pearson r koefficienter. Jeg har i ovenstående program endvidere bedt om (med tilføjelse af 'kendall') at få udskrevet Kendall tau b-koefficienter[35]. Det er en god ide indledningsvist at studere matricen med de simple korrelations-koefficienter. Pga. den store mængde udskrifter, som programmet forårsager, vises dog herunder alene resultater fra selve alpha-testen:

[35] Da der jo er tale om variabler på ordinalskala, vil det være rimeligt at studere en korrelationskoefficient, der er udviklet specielt hertil – i modsætning til Pearson r som undersøger *lineære* sammenhænge.

Correlation Analysis

Cronbach Coefficient Alpha

for RAW variables : 0.795839
for STANDARDIZED variables: 0.810855

	Raw Variables		Std. Variables	
Deleted Variabler	Correlation with Total	Alpha	Correlation with Total	Alpha
REK2V1	0.393841	0.831845	0.390087	0.834065
REK2V2	0.685489	0.725349	0.711255	0.738855
REK2V3	0.608361	0.746756	0.604005	0.772368
REK2V4	0.689518	0.723295	0.704646	0.740971
REK2V5	0.584116	0.756547	0.599255	0.773811

Vi bliver her præsenteret for to forskellige Alpha-koefficienter. Dels en koefficient for såkaldte 'RAW variables', dels for 'STANDARDIZED variables'. Som navnet siger, er der ikke blevet gjort noget ved de rå variabler, mens de andre er standardiseret til at have ens varians. Nogle gange kan der være betragtelig forskel mellem de to koefficienter, ikke mindst hvis variablerne er skaleret forskelligt (i så fald benyttes den standardiserede værdi). Minimumsværdi, maksimumsværdi og varians ved de forskellige variabler fremgår af de univariate statistikker, som udskrives i forbindelse med Alpha-beregningerne, og efter konkret, substantiel vurdering kan det vælges, hvilken Alpha-koefficient man vil bruge, samt om nogle af variablerne evt. skal rekodes inden dannelse af indeks.

I ovennævnte eksempel er variablernes varians nogenlunde ens, som også ses af de meget ens Alpha-koefficienter. Vi kan derfor nøjes med at koncentrere os om koefficienten for de rå variabler. Dette vil være langt det mest almindelige billede, når der benyttes et batteri af items med samme skala, f.eks. fra '1' til '5' som i tilfældet her. Alpha-koefficienten er på ca. 0,8, hvilket må betragtes som tilfredsstillende, da minimumskravet normalt sættes til 0,7.

Under de to koefficienter vises et skema, hvoraf fremgår de enkelte items betydning for de to Alpha-koefficienter, dels hvad koefficientens værdi ville blive, hvis pågældende item blev udeladt af indekset, dels dette items korrelation med summen (eller den standardiserede sum) af de resterende item. Bl.a. ser vi her, at variablen 'REK2V1' som den eneste ikke korrelerer stærkt med de resterende variabler (korrelation med total = 0,39), og Alpha-koefficienten stiger noget ved udtagelse af denne. Det er ikke entydigt i dette eksempel, hvorvidt variablen skal pilles ud af indekset, men det bør overvejes, hvorvidt man ud fra bl.a. en teoretisk vurdering vil have den med eller ej. Endvidere vil man i tvivlstilfælde ofte gribe det lidt praktisk an og forsøge sig med paralelle analyser - først med det ene indeks, så det næste.

9.INSIGHT - HVORDAN MAN OGSÅ KAN FORETAGE INDLEDENDE ANALYSER

Til slut skal der lige kort omtales en ganske udmærket måde, hvorpå man i kombination med de vanlige måder kan danne sig et indledende overblik over datasættet eller dele heraf, nemlig via noget der hedder INSIGHT. Her kan man meget hurtigt få vist grafer, figurer og forskellige statistiske mål - både hvad angår univariate fordelinger og bivariate sammenhænge mellem variabler. Man kommer ind i INSIGHT ved at klikke på *Globals* fra menurækken øverst på skærmen. Der viser sig nu en undermenu-bjælke, hvor der vælges *Analyse* og *Interactive data analyse*. Der skal nu vælges et databibliotek og derpå et datasæt - enten dobbeltklikkes på datasættet eller der enkeltklikkes efterfulgt af klik på *open*-knappen.

Datasættet bliver nu vist som tabel, og man kan nu klikke på *Analyse* fra menurækken for oven, hvorefter der viser sig en bjælke med forskellige analyse-muligheder. Det nemmeste vil imidlertid ofte være først at markere de variabler,der skal analyseres på. Dette gøres ved at holde Ctrl-knappen nede, mens man med museklik i de grå felter vælger de pågældende variabler. Derpå klikkes på *Analyse*, og en af funktionerne vælges. Resultaterne kommer så øjeblikkeligt frem på skærmen. Eksempelvis kan man vælge *Distribution (Y)*, hvorefter der fremkommer en række statistiske mål på de valgte variabler, og desuden bliver der vist forskellige grafiske fremstillinger. Man kan også vælge f.eks. at få stolpediagrammer/histogrammer af variabler eller scatterplots over bivariate sammenhænge mellem variabler. Ved begge valg kan man vælge en række analysevariabler, således at der vises stolpediagram-mer/histogrammer for hver enkelt variabel eller scatterplots over alle bivariate sammenhænge. Hvis man laver stolpediagrammer med variabler på nominel- eller ordinalskalaniveau, så husk blot at sørge for, at der skal stå ‘num’ i det lille grå felt over variabelnavnet. Hvis der står ‘int’, så klik en gang på feltet og vælg ‘nom’ i stedet. Ved stolpediagrammer/histogrammer kan man i øvrigt også få et første indblik i bivariate sammenhænge, for ved et museklik på en stolpe i en af de viste variabler, fremhæves ikke blot denne stolpe, men også de områder ved de øvrige variabler, hvor disse respondenter er repræsenteret. Dvs. at hvis man har valgt en variabel for køn og tre holdningsvariabler, så kan man ved f.eks. at klikke på den søjle, der hører til mændene, se hvordan mændene

har svaret på holdningsspørgsmålene. Samtidig markeres alle mændenes records (rækker) i datasættet. Og hvis man har lyst, kan man gå tilbage til datasætvisningen og trykke på højre museknap. Herved fremkommer en såkaldt *pull down* menu, og hvis man klikker på *extract*, dannes der et *subset*. Et subset er et nyt datasæt, der kun består af en del af det oprindelige. I dette tilfælde kun af mænd.

Der er masser af andre muligheder, men det vil være for omstændeligt at komme omkring det hele her. Desuden findes en udmærket INSIGHT-hjælp, hvis man vælger *Help under selection* under *Help*-menuen.

APPENDIKS - HÅNDTERING AF DATASÆT

Med hensyn til manipulation af datasæt har nærværende vejledning koncentreret sig om det, man kunne kalde for almindelige rekodninger af variabler, samt om beregning af nye variabler på baggrund af flere eksisterende variabler. Ofte vil man imidlertid have behov for helt andre typer af manipulationer på sine data, og jeg vil i dette appendiks ganske kort beskrive forskellige typer af manipulationer, der kan benyttes til løsning af nogle meget gængse problemstillinger. Det drejer sig om følgende, der vil blive beskrevet i nævnte rækkefølge:

1. Sortering af datasæt
2. Ændring af format-tilknytning
3. Tilknytning af faste SAS-formater
4. Ændringer/rettelser i enkelte records
5. Omdøbning af variable
6. Flytning af variable
7. Bevarelse af værdi i variabel fra én observation til den næste i et datastep
8. Samling af flere "sub-set" til ét samlet datasæt
9. Dannelse af "sub-set" - frasortering af records med bestemte egenskaber
10. Indlæsning af datasæt fra EXCEL
11. Hvordan man nemt laver et lille data-sæt til at afprøve programmer på
12. Eksempel på hvordan man kan teste for sammenhæng alene på baggrund af tabel-udskrifter

A.1. Sortering af datasæt

Som beskrevet i kapitel 1, kan en del manipulationer ordnes ved at klikke sig ind på et datasæt via kartoteks-ikonen - inde i "WIEV TABLE". Her kan man bl.a. sortere datasættet efter én eller flere variabler. Dette gøres ved enten at klikke på "Data" og "sort", hvorpå sorterings-variabel eller -variabler vælges, eller at klikke på en af de to sorterings-ikoner - en for stigende og en for faldende sortering. Sortering kan imidlertid også nemt

foretages med en "proc sort". F.eks. viser følgende programeksempel, hvordan datasættet "skole.elever2" sorteres efter stigende alder[36]:

```
proc sort data=SKOLE.ELEVER2 out=work.ELEVER;
  by WEIGHT;
run;
```

Læg mærke til den 'out'-option, der er indsat i første sætning. Når man benytter procedurer, der ikke blot beregner og foretager udskrift, men også manipulerer data-sættet, kan det være en fordel for en sikkerheds skyld at bevare det oprindelige datasæt. 'Out'-optionen sørger for, at sorteringen alene gælder output data-sættet. *Stigende* sortering er såkaldt default, men hvis man vil sortere i faldende orden, indsættes ordet "descending" umiddelbart før sorteringsvariablen. Og har man behov for at sortere efter flere variabler, indsættes disse efter hinanden med blanktegn imellem. Følgende programeksempel sorterer f.eks. først efter køn og derpå efter alder i faldende orden.

```
proc sort data=SKOLE.ELEVER2 out=work.ELEVER;
  by SEX descending AGE;
run;
```

A.2. Ændring af format-tilknytning

Det er også muligt at ændre format-tilknytning inde i "WIEV TABLE". Først dobbeltklikkes på det grå felt med variabel-label (eller variabel-navn hvis man har ændret visningen via "View" og "Collumn names") og derpå ændres formatnavnet. Hvis der er tale om et selvfremstillet format, skrives navnet på formatet i format-rubrikken, og hvis der er tale om et fast SAS-format, vælges dette via pilen til højre for denne rubrik. En sådan ændring kan dog også foretages ved blot at køre et "data-step"-program med en ny "format"-sætning, hvorefter en tidligere formattilknytning overskrives.

A.3. Tilknytning af faste SAS-formater

I nærværende vejledning er alene omtalt tilknytning af selvfremstillede udskrift-formater. I SAS findes imidlertid en række faste formater, der uden videre kan tilknyttes variabler. F.eks. kan det angives, hvor mange decimaler, man ønsker, at værdien til en given variabel skal udskrives med. For en fuldstændig oversigt over de mange forskellige formater henvises til SAS-manualer. Herunder gives nogle få eksempler på meget

[36] De fleste datasæt i dette appendiks benævnes blot 'ELEVER', og de indsættes alle i det midlertidige bibliotek 'WORK'. Som nævnt tidligere, er det egentlig ikke nødvendigt direkte at skrive denne biblioteksreference, da SAS automatisk regner med, at der er tale om 'WORK'-biblioteket, hvis intet andet er anført.

almindelige formattilknytninger samt en kort forklaring til hver, om hvordan variablen optræder i output.

`format var1 8.4;`	Otte cifre i alt, heraf fire decimaler
`format var2 comma9.2;`	Ni cifre i alt, heraf to decimaler, der indsættes tusindeadskiller
`format var3 commax9.2;`	Samme som ovenstående, blot dansk notation, hvor tusindeadskiller angives som '.' og decimalerne følger efter ','
`format var4 z5.;`	Fem cifre i alt, ingen decimaler, foranstillede nuller undertrykkes ikke

A.4. Ændringer/rettelser i enkelte records

Af og til vil man have behov for manuelt at ændre værdien på en variabel i en enkelt record eller at slette en hel record. Hvis man vil foretage den slags ændringer, kan dette ligeledes gøres via klik på kartoteks-ikonet og efterfølgende valg af det pågældende datasæt. Før ændringerne foretages, skal man dog ændre visnings-formen fra "Browse Mode" til "Edit Mode", hvilket gøres via klik på "Edit"-menuen. Man kan herefter frit ændre værdier, men dette er selvfølgelig noget, der skal gøres med stor varsomhed.

Noget, man oftere vil få behov for, er at slette hele records. F.eks. kan der ved en fejl være dannet nogle tomme records til slut, som man gerne vil af med. Dette gøres ved at klikke en enkelt gang på den record, der skal slettes, og derpå klikke på "Edit" og "Delete Row". Som oftest vil dette dog kunne gøres langt lettere med en simpel program-sætning inde i et "data-step". Følgende program sletter f.eks. alle records (her respondenter/elever), hvor variablen alder er mindre end '12'. Det oprindelige datasæt bevares, men det nye midlertidige datasæt "work.subset" indeholder kun elever fra 12 år og opefter.

```
data work.subset;
set SKOLE.ELEVER2;
if AGE=10 then delete;
run;
```

A.5. Omdøbning og flytning af variabler

Nogle af de meget almindelige operationer på data kan desværre ikke foretages via kartoteks-ikonet. Dette gælder f.eks. omdøbning af variabler samt flytning af variabler til et andet sted i data-matricen. Disse ting kan i stedet klares f.eks. i "wiev-table". Skriv "fswiev" efterfulgt af et blanktegn samt derpå datasættets navn i *kommando-linien* (den lille skrive-rubrik til venstre for ikonerne). Tryk derefter på ENTER, og datasættet vises. Hvis der derpå klikkes på "Edit", "Update" og "OK", kan der foretages ændringer som

via kartoteksikonen. Men derudover kan der altså også ændres i variabelnavne, og man kan flytte variabler eller serier af variabler til nye steder i matricen. Ændring i variabelnavn foregår ved at klikke på "Wiev" og "Rename", mens flytning af variabler foregår ved at klikke på "Wiew", "Arrange variables" og "Move". Ændring i variabelnavn kan dog lige så let foretages via programsætningen "rename" i et data step. F.eks. ændrer følgende program variabelnavnene "v01" og "v02" til henholdsvis "var1" og "var2".

```
data work.elever;
set skole.elever2;
rename v01=var1 v02=var2;
run;
```

A.6. Bevarelse af værdi i variabel fra en observation til den næste i et data-step

Normalt, når man foretager operationer på en variabel i et data-step, vil operationer i forbindelse med en enkelt observation ikke øve indflydelse på den følgende observation. Eller med andre ord: værdierne i variabler hænger ikke sammen fra observation til observation. Som det tidligere er beskrevet, skal man betragte et data-step som en løkke - eller i computer-terminologi et 'loop'. Og hvert gennemløb modsvares af en observation i data-sættet. Hvis således at en variabel i en regneoperation f.eks. får tildelt værdien '3', så gemmes dette tal ikke til det følgende gennemløb at programsætningerne i data-step'et. Det ville i de fleste tilfælde også være aldeles usmart, men af og til kan man få behov for at gemme værdien i en variabel fra gennemløb til gennemløb (fra observation til observation). En sådan kan kaldes for en *global* variabel. F.eks. kan man få behov for at nummerere observationerne i data-sættet fortløbende fra én og opefter. Følgende lille program laver en ny variabel 'idnr', der får en fortløbende nummerering, sådan at variablen i observation nr. 1 får værdien '01', i observation nr. 2 får værdien '02' osv.

```
data work.elever;
set skole.elever2;
idnr+1;
format idnr z3.;
run;
```

Læg først mærke til 'format'-sætningen! Den er egentlig unødvendig, men id-numre skrives typisk uden at undertrykke evt. foranstillede nuller, hvorfor det er valgt at tilknytte et format, der sørger herfor (se i afsnittet *Tilknytning af faste SAS-formater* ovenfor). Læg dernæst mærke til, at selve regneoperationen 'idnr+1' ikke svarer til den normale syntaks.

Her er vi vant til at se en sætning som f.eks. ‘IDNR=IDNR+1’. Når en ny variabel oprettes på denne måde (uden efterfølgende lighedstegn) angives hermed, at dens værdi skal gemmes fra observation til observation, samt at dens startværdi er lig med ‘0’. En anden måde at fortælle SAS, at værdien skal gemmes er ved at benytte en ‘retain’-sætning først i programmet, og ved denne metode benyttes samme syntaks som ved normal programmering i regneoperationerne. Følgende program foretager derfor nøjagtigt det samme som ovenstående.

```
data work.ELEVER;
set SKOLE.ELEVER2;
retain IDNR (0);
NYID=IDNR+1;
format IDNR z3.;
run;
```

Nullet i parantesen angiver, at startværdien i variablen ‘IDNR’ skal være ‘0’, og man kan frit skrive en anden værdi, hvis det er ønskeligt. Eksempelvis kan det tænkes, at man gerne vil have nummeret til at starte ved ‘1000’ istedet.

Til ovenstående skal knyttes en enkelt kommentar. Der findes faktisk i SAS en systemvariabel ved navn ’_n_’, som automatisk fungerer som en global variabel, og denne sætter fortløbende nummer på observationerne nøjagtigt som programmerne herover gjorde. Hvis man imidlertid ændrer rækkefølgen af observationerne og senere vil tilbage til den gamle orden, er det nødvendigt at lave sin egen nummerering.

A.7. Samling af flere “sub-set” til ét samlet datasæt

Der skal her nævnes tre hyppigt forekommende situationer, hvor man får behov for at sammenlægge flere datasæt til ét datasæt. Den ene situation opstår, når flere personer har været med i indtastningsfasen. Der er så normalt blevet oprettet lige så mange datasæt, som der har være personer med til indtastningen, og disse datasæt skal efterfølgende kædes sammen til et stort datasæt, indeholdende samtlige records - dette hedder på SAS-sprog for “concatenate”, og det kan f.eks. foretages i ASSIST, hvor man fra “Primary menu” vælger “DATA MGMT” og derpå “COMBINE” og “Concatenate”. Det er dog langt nemmere at foretage sammenkædningen via programeditoren. Følgende program-eksempel viser, hvordan to datasæt “SKOLE.DEL1” og “SKOLE.DEL2” kædes sammen til datasættet “work.ELEVER”.

```
data work.ELEVER;
set SKOLE.DEL1 SKOLE.DEL2;
run;
```

Ovenstående er et eksempel på, hvordan man føjer records sammen. Billedligt talt ligger de to del-matricer vertikalt i forlængelse af hinanden i den samlede data-matrice. Man kan imidlertid også få brug for at sammenføje del-matricer i den horisontale retning. Det kan f.eks. ske, når man arbejder med panel-data. Her har man f.eks. to datasæt bestående af de samme respondenter, men hvor svarene er fra to forskellige tidspunkter. Vi vil nu gerne have føjet disse data sammen, sådan at vi for hver respondent kan se svarene for begge tidspunkter, og vi kan foretage sammenligninger. Det er selvfølgelig nødvendigt, at der i begge datasæt eksisterer en variabel med et identifikations-nummer (gerne kaldet for id-nummer), så den samme respondent har samme nummer i begge datasæt, og sådan at der ikke forekommer gentagelser af id-numre i samme datasæt. Denne variabel skal i øvrigt have samme variabel-navn, og de øvrige variabler (eller i hvert fald en lang række af dem) har så forskellige navne. F.eks. kan variablen til spørgsmål 1 hedde "u1v1" i den første undersøgelse og "u2v1" i den anden.

En anden situation, hvor man får behov for at sammenføje matricer på den horisontale led, er hvor man har et datasæt fra en survey-undersøgelse, hvor respondenterne er indelt i klynger, f.eks. geografiske områder eller virksomheder. De enkelte klynger kan have bestemte karakteristika, som man gerne vil føje til survey-dataene. Hvis klyngerne f.eks. er kommuner, kan det dreje sig om kommunestørrelse, eller om der er tale om en land- eller bykommune. Her vil det kun være datasættet med klyngeoplysninger, hvor hver enkelt record er unik. Hvis vi igen taler om kommuner, vil hver record indeholde en variabel for kommunenummeret som identifikation samt en række kommunale oplysninger. I survey-dataene derimod vil der være mange records (respondenter) med samme kommunenummer, og ideen er, at records med samme kommunenummer skal have tilføjet de samme kommunale oplysninger.

De to ovenfor nævnte situationer, hvor vi skal have føjet datasæt sammen på den horisontale led, og hvor sammenføjningen sker via sammenligning af identifikations-variablens værdi i de to datasæt, kaldes i SAS-terminologi for "match merging". Nedenfor vises et programeksempel, hvor der til datasættet "SASDATA.SURVEY" tilføjes nogle kommunale registerdata fra datasættet "SASDATA.KOMMUNE".

```
data SASDATA.SAMLET;
merge SASDATA.SURVEY SASDATA.KOMMUNE;
  by KOMNR;
run;
```

For nærmere oplysninger om, hvordan sammenføjningen sker, kan henvises til SAS-hjælpefunktionen inden i "COMBINE"-vinduet i ASSIST-brugerfladen, og det er under alle omstændigheder nødvendigt at checke grundigt af efterfølgende, om sammen-

føjningen er sket på tilfredsstillende vis - især de første gange hvor man foretager den slags operationer.

A.8. Indlæsning af datasæt fra EXCEL eller andre programmer

Som det sidste punkt i dette apendiks vil jeg vise, hvordan man nemt indlæser data fra en EXCEL-fil. Det kan f.eks. blive aktuelt i forbindelse med tilføjelse af registerdata til et eksisterende SAS-datasæt, sådan som det blev beskrevet ovenfor - registerdata vil ofte være at finde i EXCEL-format.

Man skal huske at gemme den pågældende EXCEL-fil i en tilpas gammel version, således at den SAS-version, men benytter, kan læse den, ellers opstår der fejl under importen. Alternativt kan EXCEL-filen først gemmes som en såkaldt dbf-fil og derefter importeres. Importen foregår ved at klikke på “File” og derpå “Import”. Herefter guides man gennem import-processen, sådan at dbf-filen gemmes som et SAS datasæt.

Det importerede datasæt er nemmest at arbejde videre med, hvis variabel-*navnene* er indskrevet i den første række i EXCEL-filen - i modsat fald giver SAS selv navne til variablerne. Filer i andre formater kan importeres på samme måde.

A.9. Hvordan man nemt laver et lille data-sæt til afprøvning af programmer

Selvom man læser nok så meget i vejledninger og manualer, kan man af og til føle sig lidt usikker på, hvordan bestemte programsætninger, funktioner, aritmetiske udtryk osv., virker. Især kan det være et problem, hvis man ikke er vant til at programmere endsige tyde manualer. Selvfølgelig kan man lave det, man tror er korrekt, og så efterfølgende checke om data-sættet ser fornuftigt ud (dvs. at stikprøvecheck af forskellige variables værdier viser forventede resultater). Ofte vil det datasæt, som man sidder med, dog være stort og vanskeligt at overskue, og derfor kan det mange gange være en god ide at afprøve programstumper på et lille overskueligt prøve-data-sæt. Via programeditoren kan dette gøres ganske nemt. F.eks. opretter nedenstående program et SAS data-sæt med fire numeriske variabler og seks observationer:

```
data work.TEST1;
input v1 v2 v3 v4;
cards;
2 1  3 5
1 2 12 4
4 4  9 6
1 1 10 4
3 3  7 3
;
run;
```

Første linie skulle være kendt. I anden sætning plejer der er være en ’set’-sætning, der fortæller hvilket SAS input-sæt, der er tale om. Imidlertid er der ikke noget SAS input-sæt. Istedet dannes der et ved hjælp af en ’input’-sætning og en ’card’-sætning. ’Input’-sætningen fortæller SAS, hvilke variabler, der er i datasættet, og ’card’-sætningen fortæller, at umiddelbart efter følger en serie data-linier. Der er altså fire variabler med navnene ’v1’ til ’v4’. Når der ikke er andet specificeret ud for variabelnavnene, betyder det, at de er numeriske, og at der er blanktegn mellem variabelværdierne i de efterfølgende data-linier. Data-linierne afsluttes med et semikolon.

Det data-sæt, der således er blevet dannet, kan efterfølgende benyttes til afprøvning af forskellige data-step med aritmetiske udtryk, funktioner, eller hvad det nu er, man er usikker på. Hvis f.eks. man er usikker på, hvordan sum-funktionen og variabel-lister heri virker, kan man afprøve følgende program:

```
data TEST2;
set TEST1;
v5=sum(v1,v2,v3,v4);
v6=sum(v1-v4);
v7=sum(OF v1-v4);
v8=sum(OF v1-v3,v4);
v9=sum(v1,v3,v4);
run;
```

Alle programlinierne er syntaksmæssige korrekte. Som beskrevet i afsnit 8.1, vil variablen ’v6’ imidlertid ikke blive lig med summen af variabel-listen ’v1-v4’, men derimod lig med værdien i ’v1’ minus værdien i ’v4’. Variablerne ’v5’, ’v7’ og ’v8’ vil derimod alle blive lig med summen af variabel-listen ’v1-v4’. Foretages en proc print som følgende efter ovenstående program fås en udskrift af alle variabel-værdierne[37]:

```
proc print;
run;
```

Og udskriften ser således ud:

Obs	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	2	1	3	5	11	-3	11	11	10
2	1	2	12	4	19	-3	19	19	17
3	4	4	9	6	23	-2	23	23	19
4	1	1	10	4	16	-3	16	16	15
5	3	3	7	3	16	0	16	16	13

Har man behov for andrer typer af variabler, kan dette selvfølgelig også lade sig gøre. Nedenstående program viser f.eks., hvordan der oprettes et lille data-sæt med både numeriske og alfanumeriske variabler, og hvor en af de numeriske variabler er et decimaltal:

```
data TEST3;
input v1 v2 $6. v3 v4 2.1 v5;
cards;
01 kvinde 1  3 5
02 mand   2 12 4
03 mand   4  9 6
04 kvinde 1 10 4
05 kvinde 3  7 3
;
run;
```

Efter variablen ’v2’ er det angivet med formatet ’$6.’, at der er tale om en alfanumerisk variabel på seks karakterer, og efter variablen ’v4’ er det angivet, at der er tale om en numerisk variabel på to cifre med én decimal. For at undgå undertrykkelse af fornastillede nuller i variabel ’v1’ påhæftes SAS udskriftsformat hertil ved hjælp af en data-sætning[38]:

```
data TEST3;
set TEST3;
format v1 z2.;
run;
```

Der bedes igen om et udprint ved hjælp af en proc print:

```
proc print;
run;
```

[37] Når der i en procedure ikke skrives, hvilket data-sæt der er tale om, benyttes blot det sidst behandlede.

[38] Dette er selvfølgelig ikke nødvendigt. I øvrigt kunne man også påhæfte formatet i en proc print, men herved gælder påhæftelsen alene denne procedure, hvorimod den er permanent i et data-step.

Og udskriften ser nu således ud:

Obs	V1	V2	V3	V4	V5
1	01	kvinde	1	0.3	5
2	02	mand	2	1.2	4
3	03	mand	4	0.9	6
4	04	kvinde	1	1.0	4
5	05	kvinde	3	0.7	3

Herefter kan man så gå igang med at prøve de ting af, man er i tvivl om, hvorpå man igen printer data-sættet ud for at checke. Alternativt kan man selvfølgelig blot klikke sig ind på data-sættet, som beskrevet i afsnit 2.2.

Til slut skal kort vises, hvordan en ekstern såkaldt rådata-fil på samme vis som ovenstående interne data kan indlæses og gemmes som en SAS-fil. Jeg har i WORD gemt en fil i ren tekst-format som 'C:\DATA\RAADATA.txt'. En udskrift af filen ser ud som følger:

```
01 02 3 2
kvinde 1
02 22 1 3
mand   4
03 11 2 4
kvinde 3
04 22 3 5
kvinde 6
05 22 3 1
mand   4
06 30 2 2
kvinde 1
```

Filen består af seks observationer, som hver har seks variabler. Hver enkelt observation er delt på to linier (eller records/poster som det hedder i edb-terminologi). Jeg kunne selvfølgelig sagtens have skrevet hver observation på én linie, men for at vise, hvordan man kan indlaese observationer, der er delt op på linier, har jeg valgt at gøre det på denne vis. Følgende program indlæser dataene og gemmer dem på SAS data-sættet 'work.test4':

```
data TEST4;
infile 'd:\DATA\RAADATA.txt';
input IDNR v1 2.1 v2 v3 / v4 $6. v5;
run;
```

Infile-sætningen bruges til at indlæse eksterne data-filer. Input-sætningen bruges på samme vis som i ovenstående eksempel. Nu er der blot ikke behov for en cards-sætning. Læg mærke til hvordan der ved de variabler, der ikke er rent nummeriske uden dicimaler, skal påhæftes et format. Ved variablen 'v1' angives således, at der skal være et komma mellem de to cifre, og ved variablen 'v4' angives, at der er tale om en alfanummerisk variabel på seks tegn. Læg også mærke til hvordan det angives med en skråstreg, at der skal skiftes linie. Rådata-filen kunne have været gemt uden blanktegn mellem variablerne (der skal dog så fyldes ud med foranstillede nuller i nummeriske variabler og blanktegn i alfanummeriske variabler). Hvis denne metode benyttes, skal der imidlertid være formater til samtlige variabler, fordi SAS ikke kan vide, hvornår én variabel stopper, og en anden begynder. Efter kørsel af proc-print fåes følgende udskrift:

Obs	IDNR	V1	V2	V3	V4	V5
1	1	0.2	3	2	kvinde	1
2	2	2.2	1	3	mand	4
3	3	1.1	2	4	kvinde	3
4	4	2.2	3	5	kvinde	6
5	5	2.2	3	1	mand	4
6	6	3.0	2	2	kvinde	1

Tre hovedgrupper af forskellige indlæsninger af data er nu vist: 1) Som SAS data-sæt ved hjælp af en set-sætning; 2) som data internt i selve indlæsningsprogrammet ved hjælp input- og cards-sætninger og 3) som eksterne rådata ved hjælp af infile- og input-sætninger. Hovedvægten i vejledningen har, som før nævnt, ikke ligget på indlæsning af filer, og for yderligere information herom må derfor henvises til andet materiale (f.eks. Nielsen 1998 for indlæsning via ASSIST og Spector 1993 for indlæsning af rådata).

A.10. Eksempel på hvordan der kan testes for sammenhæng blot på baggrund af tabeludskrifter

Det blev vist i appendiks afsnit A.9, hvordan man kan indlaese data til en SAS-fil. I dette afsnit vil jeg vise et ganske nyttigt eksempel på anvendelse af et sådant lille datasæt. Man kan komme ud for situationer, hvor man ikke har tilgang til selve survey-datasættet, men kun f.eks. en eller flere tabeludskrifter fra dette. Lad os f.eks. sige, at man selv har lavet et survey, og at man samtidig har tilgang til tabeludskrifter af fordelinger fra en lignende tidligere undersøgelse. Det kunne f.eks. være tilfredshedsundersøgelser af et eller andet, hvor det samme spørgsmål er stillet i både den gamle undersøgelse og ens egen nye. Man har eksempelvis følgende frekvenser:

	Tilfreds	*Utilfreds*
Gammel survey	356	118
Ny survey	403	107

Man kan selvfølgelig beregne f.eks. Chisquare test og Gamma helt manuelt, men ofte vil det synes noget lettere at få SAS til at gøre det - især fordi den samme syntaks med få modifikationer kan benyttes i forbindelse med andre tabeller. Det kan gøres ved først at indlæse i alt fire observationer, én for hver celle, og dernæst lave en *proc freq*, hvor der benyttes en vægt, som er lig med antal observationer, sådan som det vises herunder:

```
data work.TEST;
input AAR TILFREDS VGT;
cards;
1 1 356
1 2 118
2 1 403
2 2 107
;
run;

proc freq data=TEST;
          tables AAR*TILFREDS / chisq measures;
          weight VGT;
run;
```

Indsættelsen af vægten gør, at den enkelte observation ganges op i antal med vægtværdien for denne observation. Udskriften ligner således nøjagtigt den udskrift, der ville fås, hvis vi havde haft tilgang til begge datasæt og herfra havde udskrevet krydstabel uden vægtning. I dette eksempel findes, at selvom andelen af tilfredse er steget med ca. fire procent, så er ændringen ikke statistisk signifikant.

INDEKS

fortsættes

[39] A står for Appendiks.

Afsnit

Programsætninger

Andet

LITTERATUR:

Andersen, Aage m.fl. 1995: *Brugerhåndbog i SAS*. København: Akademisk Forlag.

Bentzen m.fl. 2000: *Kompendium i kvantitativ metode*. Aalborg Universitet, Institut for Økonomi, Politik og Forvaltning,

de Vaus, D. A. 1996: *Surveys in social rsearch*. London: SAGE Publications Ltd.

Frankfort-Nachmias & David Nachmias 1997: *Research Methods in the Social sciences*. London - Sydney – Auckland: Arnold.

Goul Andersen, Jørgen 1998: *Borgerne og lovene*. Rockwool Fondens Forskningsenhed, Århus: Aarhus Universitetsforlag.

Hellevik, Ottar 1994: *Forskningsmetode i sociologi og statsvitenskab*. Oslo: Universitetsforlaget.

Hildebrand, David K. 1977: *Analysis of Ordinal Data*. Newbury Park - London - New Delhi: Sage Publications.

Kreiner, Svend 1999: *Statistisk problemløsning - Præmisser, teknik og analyse*. København: Jurist- og Økonomforbundets Forlag.

Kim, Jae-On & Charles W. Mueller 1978: *Factor Analysis - Statistical Methods and Practical Issues*. Newbury Park - London - New Delhi: Sage Publications.

Nielsen, Peter 1998: *Produktion af viden - en praktisk metodebog*. København: Teknisk Forlag.

Norusis, Marija J. 1990: *SPSS Base System User's Guide*. Chicago: SPSS Inc.

Olsen, Henning 1998: *Tallenes talende tavshed. Måleproblemer i surveyundersøgelser*. København: Akademisk Forlag A/S.

Redder, K.W., K. Siune & O., Tonsgaard 1972: *Introduktion til sociologisk metode*. København: Munksgaard.

Reid, Stuart 1987: *Working with Statistics*. Oxford: Polity Press.

Rosenberg, Moris 1968: *The Logic of Survey Analysis*. New York & London: Basic Books, Inc., Publishers.

SAS Institute 1990: *SAS® Language: Reference* (First Edition), Cary, North Carolina: SAS Institute Inc.

SAS Institute 1990: "SAS® Procedures Guide" (Third Edition), Cary, North Carolina: SAS Institute Inc.

Spector, Paul E. 1993: *SAS® Programming for Researchers and Social Scientists*. Newbury Park, London, New Delhi: Sage Publications.

SPSS Inc. 1999: *SPSS® Base 9.0 - Applications Guide*. Chicago: SPSS Inc.