



Håndbok i å lage tabeller i SAS

Februar 2007

Forord

SSB har et stort behov for å lage tabeller. Mange av tabellene vi lager blir laget i SAS av medarbeidere på fagseksjonene. Disse trenger en veiledning i hvordan vi lager tabeller med SAS. Dette er bakgrunnen for dette heftet. Vi vil her ta for oss forskjellige typer tabeller som er vanlige i SSB og beskrive hvordan man kan lage disse ved hjelp av PROC TABULATE og PROC REPORT.

For hver av prosedyrene vises først hvordan vi lager enkle tabeller. Vi tar for oss en enkel tabell først og utvider den i de påfølgende eksemplene. Videre gjennomgås hvordan vi lager prosenttabeller. Deretter tar vi for oss noen mer avanserte tabeller og teknikker. Vi viser også hvordan vi får tabeller ut til html-filer.

Dette heftet er en utvidet og oppdatert versjon av Interne dokumenter 2001/1. Nyheter i sas versjon 9.1 er tatt inn i heftet og den viktigste nyheten er at det er lettere å lage ikke-hierarkiske tabeller nå enn tidligere. Dessuten beskriver vi nå også hvordan vi kan lage tabeller med PROC REPORT.

Håndboken er skrevet av Kristian Lønø ved IT-seksjonen på avdeling for økonomisk statistikk. Forfatteren takker Liv Daasvatn for godt samarbeid i arbeidet med den første utgaven av heftet og for inspirasjon til denne håndboken.

Statistisk sentralbyrå,
Oslo, 8. februar 2007

Øystein Olsen

Innhold

Innledning	3
Kort om syntaksen i TABULATE.....	4
Hvordan lage enkle tabeller med TABULATE.....	5
Prosenttabeller.....	10
Prosent basert på summen av en analysevariabel, PCTSUM.....	10
Kolonnesprosent basert på summen av en analysevariabel, COLPCTSUM	11
Linjesprosent basert på summen av en analysevariabel, ROWPCTSUM	12
Kolonnesprosent basert på antall observasjoner, COLPCTN.....	13
Flere variable i forspalten	14
Delprosent med flere variable i forspalten.....	15
Prosent av en annen variabel.....	16
Hvordan lage mer avanserte tabeller i TABULATE	17
Penere utseende på tallene i tabellen	17
Tabell med flere grupperingsvariable og flere analysevariable	18
Samme variabel til både gruppering og analyse.....	20
Analysevariabel i hodet og statistikk nøkkelord i forspalten	21
Veiing, oppblåsing.....	22
Gjøre om en grupperingsvariabel til en analysevariabel	25
Ratetabeller	28
Ikke-hierarkiske tabeller	29
TABULATE kontra andre PROC'er	32
TABULATE istedenfor PROC MEANS	32
TABULATE istedenfor PROC FREQ.....	34
Manglende verdier, missing	35
Manglende verdier i grupperingsvariable	35
Manglende verdier i analysevariable	37
Utfyllende om syntaksen i TABULATE	38
Feilmeldinger i TABULATE	47
Finner ikke variabelen som brukes i tabellen.....	47
Statistiske beregninger uten analysevariabel.....	47
Statistiske beregninger i både forspalte og hode.....	47
Ufullstendig nevner i prosentuering.....	48
Ingen data å lage tabell av.....	48
Kort om syntaksen i REPORT	49
Hvordan lage enkle tabeller med REPORT.....	50
Hvordan lage mer avanserte tabeller i REPORT.....	54
Beregne nye kolonner	54
Flere nivåer i forspalten	57
Lage en sammensatt forspalte ut fra to grupperinger	58
Utfyllende om syntaksen til REPORT.....	62
Feilmeldinger i REPORT	69
Statistiske beregninger uten analysevariabel.....	69
Flere statistiske beregninger knyttet sammen samtidig	70
Ingen data å lage tabell av.....	70
Tabeller skrevet direkte til html-fil.....	71
Utskrift fra SAS til Excel eller Word	72
Behandle unix-filer fra windows	72
Definisjoner av statistiske nøkkelord	74
Vedlegg	75
Innlesingsprogram for befolkningsdatafilen bef	75
Innlesingsprogram for arbeidskraftundersøkelsens (AKU) datafil aku	76
Programmet som lager tekstformater som brukes	77
Stikkordregister.....	78

Innledning

Først: Hva menes med en tabell? I det følgende vil vi benytte disse begrepene når vi beskriver en tabell: tabelloverskrift, forspalte, hode, forspaltehode, tabellinnmat, kolonne, linje og celle.

Tabelloverskrift		Kolonne			
Antall hotellsenger etter hotelltype og fylke.					
Forspaltehode	Fylke	Byhotell	Landhotell	Turist- og høyfjells-hotell	Hode
Forspalte	I alt	39357	47959	25646	Tabell-innmat
Østfold	1204	597	-	-	
Akershus	-	3632	-	-	
Nordland	1756	2473	358		
Troms	2033	1363	90		
Finnmark	512	1960	450		
Linje					Celle

PROC TABULATE (heretter kalt TABULATE) er SAS' spesialverktøy for å lage tabeller. Med utgangspunkt i et SAS datasett kan vi lage tabeller i opptil 3 dimensjoner (side, forspalte og hode). Det vanligste er likevel 2 dimensjoner: forspalte og hode (eller om man vil: linje og kolonne). Resultatet fra TABULATE er en ferdig tabell som legges enten i utskriftsvinduet i SAS eller på en fil (altså ikke et SAS datasett).

SAS datasettets variable brukes i TABULATE enten til gruppering eller klassifisering (variablene omtales som CLASS-variable), eller til analyse (variablene omtales som VAR-variable). Variable kan også brukes til veiing (WEIGHT). Grupperingsvariablene danner forspalten og/eller hodet, mens analysevariablene utgjør selve tabellinnmaten i form av summer, gjennomsnitt, standard avvik, etc. Ved hjelp av TABULATE kan vi lage alt fra enkle til relativt komplekse tabeller med korte programmer.

PROC REPORT (heretter kalt REPORT) er en prosedyre som lager rapporter. Disse kan til forveksling ligne tabeller og vi kan derfor bruke REPORT til å lage en del typer tabeller.

Programeksempler er vist slik:

```
PROC TABULATE DATA=bef;
```

Eksempler på tabeller er vist slik:

```
Tabell 1   Befolkning
```

Frivillige valg (*options*) er plassert slik:

```
<frivillig valg>
```

Lure tips er markert med en lyspære i margen

Fallgruver og andre uhumskheter er markert med en bombe i margen

Mer om TABULATE og REPORT finnes i det interaktive hjelpesystemet i SAS og på SAS sine nettsider: <http://support.sas.com/documentation/index.html>. Det er også et egne kapitler om TABULATE og REPORT i Håndbok i SAS, del 2: Oppslag (SSH 71).

Kort om syntaksen i TABULATE

Dette er den vanligste oppbygningen av et TABULATE-program:

```
PROC TABULATE DATA= SAS datasett FC='-----' NOSEPS MISSING ;  
  CLASS grupperingsvariable ;  
  VAR analysevariable ;  
  TABLE tabellspesifikasjon ;  
  TITLE 'overskrift' ;
```

Programmet forklares slik:

PROC TABULATE-setningen:

I denne setningen oppgir vi hvilket SAS datasett som skal være utgangspunkt for tabellen (**DATA=**). Videre sier vi hvilke tegn som skal brukes i tabellens ramme (**FC=**). Ved å bruke **FC=** som over (1 blank og 10 streker) sammen med **NOSEPS**, vil tabellen bare ha vannrette streker i rammen. Dette gir tabellen en grei utforming. **MISSING** betyr at observasjoner med manglende verdier (missing) i grupperingsvariablene tas med i tabellen.

CLASS-setningen:

I **CLASS** lister vi opp grupperingsvariablene med en blank mellom hver (eks: fylke kjønn etc).

VAR -setningen:

I **VAR** lister vi opp analysevariablene, som *må* være numeriske (eks: alder inntekt etc).

TABLE-setningen:

Med **TABLE** beskrives tabellen. Her angis de forskjellige tabelldimensjoner (side, forspalte og hode) med tilhørende variable, samt hvilke statistiske beregninger som skal utføres. Variable som brukes i tabellen *må* være definert enten i **CLASS** eller i **VAR**-setningen.

Tabeller kan lages i opptil 3 dimensjoner: side, forspalte og hode. Side, forspalte og hode spesifiseres på samme måte og skilles fra hverandre med komma. Spesifikasjonene kan inneholde følgende elementer:

grupperingsvariable	Definert i CLASS-setningen
analysevariable	Definert i VAR-setningen
ALL	Egen grupperingsvariabel for totaler
statistiske nøkkelord	MEAN, MAX, MIN, N, PCTN, PCTSUM, STD, SUM etc.
formater	Bestemmer tallenes utseende og kolonnebredde (antall desimaler etc.)
tekst	Erstatter variabelnavn eller statistiske nøkkelord med annen tekst

Følgende skilletegn kan brukes i TABLE-setningen:

,	komma	skiller side-, forspalte- og hode-spesifikasjonene
	blank	setter elementer ved siden av hverandre i hodet og nedenfor hverandre i forspalten
*	stjerne	krysser elementer, setter elementer 'innenfor' hverandre
()	parenteser	grupperer elementer

Fortvil ikke, om betegnelsene ovenfor virker kryptiske. Eksempler forklarer bedre enn mange ord!

Hvordan lage enkle tabeller med TABULATE

I det følgende vil vi skritt for skritt ta for oss tabeller som først er helt enkle, men som gradvis forfines.

De første eksemplene benytter et datasett med befolkningstall. Beskrivelse av filen står i vedlegg på side 75. Variable på datasettet er blant annet årstall (variabel: aar), kjønn (variabel: kjønn), antall innbyggere pr. 1.1 (variabel: innbyggere), antall døde (variabel: dode), antall innvandrede (variabel: innvandrede), etc.

Det første eksempelet viser en tabell med én grupperingsvariabel (landsdel) og én analysevariabel (innbyggere). For grupperingsvariable lages det en linje (ev. en kolonne) for hver verdi variabelen har på SAS datasettet det lages tabell fra. Analysevariable må være numeriske og de brukes til beregninger, f.eks. summering. Kommaet i tabellsetningen angir at landsdel skal utgjøre forspalten, mens antall innbyggere plasseres i hodet. Når vi ikke angir noe statistikk-nøkkelord, blir analysevariabelen summert. (Statistikk-nøkkelordet SUM benyttes altså på analysevariable hvis vi ikke ber om noe annet.). Vi tar med MISSING-option for å sørge for at manglende verdier for grupperingsvariable (her: landsdel) kommer ut i tabellen som egne kategorier.

```
PROC TABULATE DATA=bef FC='-----' NOSEPS MISSING;
  CLASS landsdel ;
  VAR innbyggere ;
  TABLE landsdel
         ,
         innbyggere
  ;
  TITLE 'Tabell 1. Antall innbyggere etter landsdel' ;
RUN;
```

Slik blir denne tabellen:

Tabell 1. Antall innbyggere etter landsdel

	Innbyggere pr. 1.1
	Sum
Landsdel	
1	4358816.00
2	1242590.00
3	2080685.00
4	1180837.00
9	.

I det forrige eksempelet var tabellsetningen veldig enkel, med kun én grupperingsvariabel og én analysevariabel. Nå utvider vi programmet litt: Vi vil ha med totalen (ALL) for alle år. Dessuten vil vi fordele innbyggerne etter kjønn. Det betyr at vi må ha med en grupperingsvariabel til, nemlig kjønn. (Variabelen kjønn må altså være med i CLASS-setningen.) For å fordele innbyggere etter kjønn plasserer vi en * mellom grupperingsvariabelen (kjønn) og analysevariabelen (innbyggere). Vi tar også med en kolonne som inneholder antall observasjoner i datasettet (N):

```
PROC TABULATE DATA=bef FC='-----' NOSEPS MISSING;
  CLASS landsdel kjønn;
  VAR innbyggere ;
  TABLE ALL landsdel
         ,
         kjønn*innbyggere N
  ;
  TITLE 'Tabell 2. Antall innbyggere etter kjønn og landsdel' ;
RUN;
```

Slik blir denne tabellen:

Tabell 2. Antall innbyggere etter kjønn og landsdel

	Kjønn			N
	1		2	
	Innbyggere pr. 1.1	Innbyggere pr. 1.1	Innbyggere pr. 1.1	
	Sum	Sum	Sum	
All	.	4384665.00	4478263.00	50.00
Landsdel				
1	.	2137548.00	2221268.00	12.00
2	.	618463.00	624127.00	12.00
3	.	1035805.00	1044880.00	12.00
4	.	592849.00	587988.00	12.00
9	.	.	.	2.00

Fremdeles er tabellsetningen enkel, så tabellen er ikke spesielt pen. I neste eksempel utvider vi derfor programmet litt mer.

Vi vil fjerne variabelnavn og statistikk-nøkkelord fra tabellen. Det gjør vi ved å sette '=' etter variabelnavn (kjønn, innbyggere) og statistikk-nøkkelord. Det er dette vi mener med å ha *tekst* med i tabellsetningen. Teksten er i dette tilfelle blank, *ingenting*. På totallinjen (ALL) vil vi ha teksten 'I alt'. Vi vil ikke ha med desimaler i antallet senger, derfor angir vi i selve PROC TABULATE-setningen et standard format på 8 tegn totalt, ingen desimaler (F=8.). Formater kan også angis i selve TABLE-setningen.

Kjønn 1 eller 2 sier ingenting om hvilket kjønn det dreier seg om. For å erstatte variabelverdiene med forståelig tekst, har vi derfor med FORMAT-setningen der vi kobler grupperingsvariabelen til formater har definert tidligere med PROC FORMAT.

```

PROC FORMAT ;
  VALUE $kjonn
    '1'  = 'Menn'
    '2'  = 'Kvinner'
    OTHER = 'Uoppgitt'
  ;
  VALUE $ldel
    '1'  = 'Øst'
    '2'  = 'Sør'
    '3'  = 'Vest'
    '4'  = 'Nord'
    OTHER = 'Uoppgitt'
  ;
RUN;
PROC TABULATE DATA=bef FC='-----' NOSEPS F=8. MISSING;
  CLASS landsdel kjonn;
  VAR innbyggere ;
  TABLE ALL='I alt' landsdel=' '
    ,
    kjonn=' '*innbyggere=' '*SUM=' ' N
  ;
  FORMAT kjonn $kjonn.
  landsdel $ldel.
  ;
  TITLE 'Tabell 3. Antall innbyggere etter kjønn og landsdel' ;
RUN;

```

Slik blir denne tabellen:

Tabell 3. Antall innbyggere etter kjønn og landsdel

	Uoppgitt	Menn	Kvinner	N
I alt	.	4384665	4478263	50
Øst	.	2137548	2221268	12
Sør	.	618463	624127	12
Vest	.	1035805	1044880	12
Nord	.	592849	587988	12
Uoppgitt	.	.	.	2

Vi ønsker å lage en kolonne for begge kjønn, i tillegg til kolonnene pr. kjønn. Dette gjør vi ved å legge til ALL i hodedimensjonen og knytte den opp mot variabelen innbyggere. SSB benytter bindestrek (-) som tegn for manglende tall i tabellen. Det får vi til med MISSTEXT='-' som *option* (frivillig valg) bakerst i tabellsetningen. For at tabellen ikke skal bli altfor bred, benytter vi en annen opsjon til tabellsetningen, nemlig bredden på forspalten (RTS, row title space). Vi setter forspaltebredden til 12 tegn (RTS=12). Vi ønsker teksten 'Landsdel' i forspaltehodet og det gjør vi med BOX-option.

```
PROC TABULATE DATA=bef FC='-----' NOSEPS F=8. MISSING;
  CLASS landsdel kjønn;
  VAR innbyggere ;
  TABLE ALL='I alt' landsdel=' '
    ,
    (ALL='Begge kjønn' kjønn=' ')*innbyggere=' '*SUM=' ' N
    / RTS=12 MISSTEXT='-' BOX='Landsdel'
  ;
  FORMAT kjønn $kjønn.;
  TITLE 'Tabell 4. Antall innbyggere etter kjønn og landsdel' ;
RUN;
```

Nå er tabellen blitt ganske bra:

Tabell 4. Antall innbyggere etter kjønn og landsdel

Landsdel	Begge kjønn	Uoppgitt	Menn	Kvinner	N
I alt	8862928	-	4384665	4478263	50
Øst	4358816	-	2137548	2221268	12
Sør	1242590	-	618463	624127	12
Vest	2080685	-	1035805	1044880	12
Nord	1180837	-	592849	587988	12
Uoppgitt	-	-	-	-	2

Hvis vi selv ønsker å bestemme rekkefølgen for grupperingsvariable, må vi først endre litt på formatene som lages. Vi må legge til NOTSORTED-option i VALUE-setningen i PROC FORMAT. Vi må også skrive tekstene i den rekkefølgen vi ønsker. Så må vi legge til option ORDER=data i PROC TABULATE-setningen og til slutt legge inn PRELOADFMT-option til CLASS-setningen.

```

PROC FORMAT ;
  VALUE $kjonn (NOTSORTED)
    '2' = 'Kvinner'
    '1' = 'Menn'
    OTHER = 'Uoppgitt'
;
RUN;

PROC FORMAT ;
  VALUE $ldel (NOTSORTED)
    '4' = 'Nord'
    '3' = 'Vest'
    '1' = 'Øst'
    '2' = 'Sør'
    OTHER = 'Uoppgitt'
;
RUN;

PROC TABULATE DATA=bef FC='-----' NOSEPS F=8. MISSING ORDER=DATA;
  CLASS kjonn landsdel /PRELOADFMT;
  VAR innbyggere ;
  TABLE ALL='I alt' landsdel=' '
    ,
    (ALL='Begge kjønn' kjonn=' ')*innbyggere=' '*SUM=' ' N
    / RTS=12 MISSTEXT='- ' BOX='Landsdel'
;
  FORMAT kjonn $kjonn.
         landsdel $ldel.
;
  TITLE 'Tabell 5. Antall innbyggere etter kjønn og landsdel' ;
RUN;

```

Nå har vi selv bestemt rekkefølgen for linjer og kolonner:

Tabell 5. Antall innbyggere etter kjønn og landsdel

Landsdel	Begge kjønn	Kvinner	Menn	Uoppgitt	N
I alt	8862928	4478263	4384665	-	50
Nord	1180837	587988	592849	-	12
Vest	2080685	1044880	1035805	-	12
Øst	4358816	2221268	2137548	-	12
Sør	1242590	624127	618463	-	12
Uoppgitt	-	-	-	-	2

De tabellene vi hittil har laget er todimensjonale, med forspalte og hode som de to dimensjonene. Vi kan utvide til 3 dimensjoner. Vi kan f.eks. legge landsdel i forspalten istedenfor årstall. Årstall kan vi så flytte til sidedimensjonen. Det vil da lages en tabell pr. årstall, med landsdel i forspalten og kjønn i hodet.

```

PROC TABULATE DATA=bef FC=' -----' NOSEPS F=8. MISSING ORDER=DATA;
  CLASS aar kjønn landsdel /PRELOADFMT;
  VAR innbyggere ;
  TABLE aar=' '
        ,
        ALL='I alt' landsdel=' '
        ,
        (ALL='Begge kjønn' kjønn=' ')*innbyggere=' '*SUM=' ' N
        / RTS=12 MISSTEXT='- ' BOX='Landsdel'
  ;
  FORMAT kjønn $kjønn.
         landsdel $ldel.
  ;
  TITLE 'Tabell 6. Antall innbyggere etter kjønn og landsdel' ;
RUN;

```

Det lages nå to tabeller:

Tabell 6. Antall innbyggere etter kjønn og landsdel

1998

Landsdel	Begge kjønn	Kvinner	Menn	Uoppgitt	N
I alt	4417599	2232493	2185106	-	25
Nord	591232	294359	296873	-	6
Vest	1037762	521174	516588	-	6
Øst	2170559	1106388	1064171	-	6
Sør	618046	310572	307474	-	6
Uoppgitt	-	-	-	-	1

Tabell 6. Antall innbyggere etter kjønn og landsdel

1999

Landsdel	Begge kjønn	Kvinner	Menn	Uoppgitt	N
I alt	4445329	2245770	2199559	-	25
Nord	589605	293629	295976	-	6
Vest	1042923	523706	519217	-	6
Øst	2188257	1114880	1073377	-	6
Sør	624544	313555	310989	-	6
Uoppgitt	-	-	-	-	1

Prosenttabeller

Å lage *enkle* prosenttabeller i TABULATE er lett, men hvis prosenteringen skal være kompleks, blir tabellene også vanskeligere å programmere. Det første vi må finne ut, er hvilken vei prosenteringen skal gå. Altså om hver enkelt *kolonne* skal summeres opp til 100 prosent (prosentuere *nedover*), eller om hver enkelt *linje* skal summeres opp til 100 prosent (prosentuere *bortover*). En tredje mulighet er å prosentueres av *totalen*. Da vil hvert enkelt tall i tabellen regnes i prosent av totalen (*grand total*) og det er *samtlig*e celler i tabellen som summerer seg opp til 100 prosent. Det er dette siste SAS bruker som standard, men det er sjelden slik prosenttering brukes i SSBs tabeller.

For å styre hvilken vei prosenteringen skal gå (nedover eller bortover), velger vi prefiks til PCTN eller PCTSUM. COL som prefiks gir kolonneprosenter og ROW gir linjeprosenter. Vi har også en mulighet til å oppgi hva som skal være nevner i prosentteringsbrøken (denominator). Nevneren oppgis som variabelnavn, enten alene eller i kombinasjon med ALL. Nevneren må omgis av < >. Hvis vi ikke oppgir noe prefiks eller en nevner, brukes standard prosenttering. Vi kan regne prosent enten av antall observasjoner (PCTN) eller av en variabels summerte verdier (PCTSUM).

Prosentter basert på summen av en analysevariabel, PCTSUM

La oss først se på et eksempel med SAS standard prosenttering. Vi vil regne prosent på grunnlag av verdien til variabelen innbyggere, og prosentene skal være fordelt på landsdel og kjønn. (Vi regner bare med 1999, derfor WHERE-setningen.). Analysevariabelen må defineres i VAR-setningen og må også brukes i TABLE-setningen. Forøvrig konstrueres TABLE-setningen på samme måte, uansett om vi benytter PCTSUM eller PCTN.

```
PROC TABULATE DATA=bef FC='-----' NOSEPS MISSING ORDER=DATA;
  WHERE aar = '1999';
  CLASS kjønn landsdel /PRELOADFMT;
  VAR innbyggere ;
  TABLE ALL='I alt' landsdel=' '
        '
        (ALL='Begge kjønn' kjønn=' ')*innbyggere=' '*PCTSUM=' '*F=numx9.1
        / RTS=12 MISSTEXT='- ' BOX='Landsdel'
        ;
  FORMAT kjønn $kjønn.
         landsdel $ldel.
        ;
  TITLE 'Tabell 7. Innbyggere etter kjønn og landsdel. Prosent av total' ;
RUN;
```

Slik blir denne tabellen. Legg merke til at prosentene regnes av totalen. Dette fordi vi har brukt PCTSUM.

Tabell 7. Innbyggere etter kjønn og landsdel. Prosent av total

Landsdel	Begge kjønn	Kvinner	Menn	Uoppgitt
I alt	100,0	50,5	49,5	-
Nord	13,3	6,6	6,7	-
Vest	23,5	11,8	11,7	-
Øst	49,2	25,1	24,1	-
Sør	14,0	7,1	7,0	-
Uoppgitt	-	-	-	-

Kolonneprosenter basert på summen av en analysevariabel, COLPCTSUM

Neste eksempel viser hvordan vi lager kolonneprosenter. Vi bruker COLPCTSUM for å angi at hver kolonne skal summere seg til 100.

```

PROC TABULATE DATA=bef FC='-----' NOSEPS MISSING ORDER=DATA;
  WHERE aar = '1999';
  CLASS kjonn landsdel /PRELOADFMT;
  VAR innbyggere ;
  TABLE ALL='I alt' landsdel=' '
    ,
    (ALL='Begge kjønn' kjonn=' ')*
    innbyggere=' '*COLPCTSUM=' '*F=numx9.1
    / RTS=12 MISSTEXT='- ' BOX='Landsdel'
  ;
  FORMAT kjonn $kjonn.
         landsdel $ldel.
  ;
  TITLE 'Tabell 8. Innbyggere etter kjønn og landsdel. Kolonneprosent' ;
RUN;

```

Slik blir tabellen:

Tabell 8. Innbyggere etter kjønn og landsdel. Kolonneprosent

Landsdel	Begge kjønn	Kvinner	Menn	Uoppgitt
I alt	100,0	100,0	100,0	-
Nord	13,3	13,1	13,5	-
Vest	23,5	23,3	23,6	-
Øst	49,2	49,6	48,8	-
Sør	14,0	14,0	14,1	-
Uoppgitt	-	-	-	-

Linjeprosenter basert på summen av en analysevariabel, ROWPCTSUM

I neste eksempel prosentuerer vi *bortover*. Vi bruker ROWPCTSUM for å angi at hver linje skal summere seg til 100.

```

PROC TABULATE DATA=bef FC='-----' NOSEPS MISSING ORDER=DATA;
  WHERE aar = '1999';
  CLASS kjonn landsdel /PRELOADFMT;
  VAR innbyggere ;
  TABLE ALL='I alt' landsdel=' '
    (ALL='Begge kjønn' kjonn=' ')*
    innbyggere=' '*ROWPCTSUM=' '*F=numx9.1
    / RTS=12 MISSTEXT='- ' BOX='Landsdel'
  ;
  FORMAT kjonn $kjonn.
         landsdel $ldel.
  ;
  TITLE 'Tabell 9. Innbyggere etter kjønn og landsdel. Linjeprosent' ;
RUN;

```

Slik blir tabellen:

Tabell 9. Innbyggere etter kjønn og landsdel. Linjeprosent

Landsdel	Begge kjønn	Kvinner	Menn	Uoppgitt
I alt	100,0	50,5	49,5	-
Nord	100,0	49,8	50,2	-
Vest	100,0	50,2	49,8	-
Øst	100,0	50,9	49,1	-
Sør	100,0	50,2	49,8	-
Uoppgitt	-	-	-	-

Kolonneprosjenter basert på antall observasjoner, COLPCTN

Hittil har vi kun vist prosentregning av *summen av analysevariabelen innbyggerer*, PCTSUM. Nå skal vi ta for oss PCTN, der prosentene regnes av *antall observasjoner i datasettet*. Når vi regner prosentene av antall observasjoner, trenger vi ingen analysevariabel, da holder det bare å ha med grupperingsvariable. Analysevariabelen må også være med i TABLE-setningen. Forøvrig konstrueres TABLE-setningen på samme måte, uansett om vi benytter PCTSUM eller PCTN.

Vi tar for oss det siste eksempelet igjen, men nå vil vi regne prosentene av antall *observasjoner i datasettet*, ikke antall innbyggere:

```

PROC TABULATE DATA=bef FC='-----' NOSEPS MISSING ORDER=DATA;
  WHERE aar = '1999';
  CLASS kjonn landsdel /PRELOADFMT;
  TABLE ALL='I alt' landsdel=' '
        ,
        (ALL='Begge kjønn' kjonn=' ')*COLPCTN=' '*F=numx9.1
        / RTS=12 MISSTEXT='- ' BOX='Landsdel'
  ;
  FORMAT kjonn      $kjonn.
         landsdel  $ldel.
  ;
  TITLE 'Tabell 10. Antall observasjoner etter kjønn og landsdel. Prosent' ;
RUN;

```

Slik blir tabellen:

Tabell 10. Antall observasjoner etter kjønn og landsdel. Prosent

Landsdel	Begge kjønn	Kvinner	Menn	Uoppgitt
I alt	100,0	100,0	100,0	100,0
Nord	24,0	25,0	25,0	-
Vest	24,0	25,0	25,0	-
Øst	24,0	25,0	25,0	-
Sør	24,0	25,0	25,0	-
Uoppgitt	4,0	-	-	100,0

Flere variable i forspalten

I de neste eksemplene vil vi lage prosenttabeller med mer enn én variabel i forspalten. Nå vil vi at landsdel og kjønn skal utgjøre forspalten, mens aldersgruppe utgjør hodet. Vi bytter altså litt rundt på forspalten og hodet og legger til et nivå i forspalten. Videre tar vi bare med gyldige koder for landsdel. Vi lager kolonnepresentert av summen av innbyggere:

```

PROC FORMAT ;
  VALUE $aldgrp
    '1'  = '0-19 år'
    '2'  = '20-67 år'
    '3'  = '67 år og over'
    OTHER = 'Uoppgitt'
;
RUN;
PROC TABULATE DATA=bef FC='-----' NOSEPS MISSING ORDER=DATA;
  WHERE aar = '1999' AND landsdel IN('1','2','3','4');
  CLASS aldersgruppe kjonn landsdel /PRELOADFMT;
  VAR innbyggere ;
  TABLE ALL='I alt' landsdel=' *kjonn=' '
    ,
    (ALL='I alt' aldersgruppe=' ')*
    innbyggere=' *COLPCTSUM=' '*F=numx9.1
    / RTS=22 MISSTEXT='- ' BOX='Landsdel og kjønn'
;
  FORMAT kjonn          $kjonn.
         landsdel      $ldel.
         aldersgruppe  $aldgrp.
;
  TITLE 'Tabell 11. Innbyggere etter kjønn, landsdel og alder. Prosent' ;
RUN;

```

Legg merke til at hver kolonne summerer seg til 100:

Tabell 11. Innbyggere etter kjønn, landsdel og alder. Prosent

Landsdel og kjønn		I alt	0-19 år	20-67 år	67 år og over
I alt		100,0	100,0	100,0	100,0
Nord	Kvinner	6,6	6,6	6,4	7,7
	Menn	6,7	7,0	6,8	5,4
Vest	Kvinner	11,8	11,9	11,2	14,1
	Menn	11,7	12,5	11,8	9,7
Øst	Kvinner	25,1	22,5	25,0	30,3
	Menn	24,1	23,7	25,2	20,3
Sør	Kvinner	7,1	7,6	6,7	7,4
	Menn	7,0	8,0	7,0	5,0

Delprosjenter med flere variable i forspalten

I neste eksempel vil vi også kolonneprosjenter, men nå ønsker vi at hver landsdel skal summerer seg til hundre. Først utvider vi tabellen med å få inn summer for begge kjønn. Videre ønsker vi også å få totalen kjønnsfordelt. Selve prosenteringen angir vi på en annen måte enn tidligere. Vi setter uttrykket vi ønsker å regne prosentene av i klammer bestående av <>. Nevneren i prosenteringen vil her bli <kjonn ALL>, med kjønn først. Dette fordi det er en regel i SAS at det mest aggregerte, i dette tilfellet ALL, må stå *sist* i nevneruttrykket.

```

PROC TABULATE DATA=bef FC=' -----' NOSEPS MISSING ORDER=DATA;
  WHERE aar = '1999' AND landsdel IN('1','2','3','4');
  CLASS aldersgruppe kjønn landsdel /PRELOADFMT;
  VAR innbyggere ;
  TABLE (ALL='I alt' landsdel=' ')*(ALL='Begge kjønn' kjønn=' ')
    ,
    (ALL='I alt' aldersgruppe=' ')*
    innbyggere=' '*PCTSUM<kjonn ALL>=' '*F=numx9.1
    / RTS=27 MISSTEXT='- ' BOX='Landsdel og kjønn'
  ;
  FORMAT kjønn          $kjonn.
         landsdel       $ldel.
         aldersgruppe   $aldgrp.
  ;
  TITLE 'Tabell 12. Innbyggere etter kjønn, landsdel og alder. Prosent' ;
RUN;

```

Slik blir tabellen:

Tabell 12. Innbyggere etter kjønn, landsdel og alder. Prosent

Landsdel og kjønn		I alt	0-19 år	20-67 år	67 år og over
I alt	Begge kjønn	100,0	100,0	100,0	100,0
	Kvinner	50,5	48,7	49,3	59,6
	Menn	49,5	51,3	50,7	40,4
Nord	Begge kjønn	100,0	100,0	100,0	100,0
	Kvinner	49,8	48,6	48,4	58,6
	Menn	50,2	51,4	51,6	41,4
Vest	Begge kjønn	100,0	100,0	100,0	100,0
	Kvinner	50,2	48,7	48,9	59,3
	Menn	49,8	51,3	51,1	40,7
Øst	Begge kjønn	100,0	100,0	100,0	100,0
	Kvinner	50,9	48,7	49,9	59,9
	Menn	49,1	51,3	50,1	40,1
Sør	Begge kjønn	100,0	100,0	100,0	100,0
	Kvinner	50,2	48,7	49,1	59,8
	Menn	49,8	51,3	50,9	40,2

Prosent av en annen variabel

Vi ønsker nå å finne ut hvor mange prosent av befolkningen som dør og hvor mange prosent som innvandrer. Vi bruker da analysevariablene dode og innvandrede. I prosentueringen bruker vi innbyggere som nevner, det gjør at vi får beregnet henholdsvis døde og innvandrede i prosent av antall innbyggere.

```

PROC TABULATE DATA=bef FC='-----' NOSEPS MISSING ORDER=DATA;
  WHERE aar = '1999' AND landsdel IN('1','2','3','4');
  CLASS kjonn landsdel /PRELOADFMT;
  VAR innbyggere dode innvandrede;
  TABLE (ALL='I alt' landsdel=' ')*(ALL='Begge kjønn' kjonn=' ')
    ,
    innbyggere='Innbyggere'*SUM=' '*F=11.
    dode*SUM=' '*F=8.
    innvandrede*SUM=' '*F=11.
    ALL='I prosent av innbyggere'*
    (dode*PCTSUM<innbyggere>=' '*F=numx9.2
    innvandrede*PCTSUM<innbyggere>=' '*F=numx11.2)
    / RTS=25 MISSTEXT='- ' BOX='Landsdel og kjønn'
  ;
FORMAT kjonn          $kjonn.
        landsdel      $ldel.
        aldersgruppe  $aldgrp.
  ;
TITLE 'Tabell 13. Innbyggere, døde og innvandrede. Landsdel og kjønn.' ;
RUN;

```

Tabell 13. Innbyggere, døde og innvandrede. Landsdel og kjønn.

Landsdel og kjønn		I prosent av innbyggere				
		Innbyggere	Døde	Innvandrede	Døde	Innvandrede
I alt	Begge kjønn	4445329	45170	41841	1,02	0,94
	Kvinner	2245770	22744	20977	1,01	0,93
	Menn	2199559	22426	20864	1,02	0,95
Nord	Begge kjønn	589605	5950	5618	1,01	0,95
	Kvinner	293629	2890	2753	0,98	0,94
	Menn	295976	3060	2865	1,03	0,97
Vest	Begge kjønn	1042923	10312	7593	0,99	0,73
	Kvinner	523706	5089	3674	0,97	0,70
	Menn	519217	5223	3919	1,01	0,75
Øst	Begge kjønn	2188257	23356	22328	1,07	1,02
	Kvinner	1114880	12045	11447	1,08	1,03
	Menn	1073377	11311	10881	1,05	1,01
Sør	Begge kjønn	624544	5552	6302	0,89	1,01
	Kvinner	313555	2720	3103	0,87	0,99
	Menn	310989	2832	3199	0,91	1,03

Hvordan lage mer avanserte tabeller i TABULATE

I dette kapittelet skal vi ta for oss noen mer avanserte tabeller. Vi stiller nå større krav til tabellens innhold og utseende. Først litt om hvordan vi endrer tallenes utseende slik at de passer bedre med SSBs standard.

Penere utseende på tallene i tabellen

Tallene i SSB-tabeller skal følge visse regler: De skal ha skilletegn (blank) mellom hver tusen (såkalt *tusenskilte*, *tusenåpning*), tall mindre enn 0,5 av den brukte enheten, skal angis med 0 og 0 selv skal angis med bindestrek (-). Dette får vi til ved å bruke formater som spesiallaget for SSBs krav. Formatene er tilgjengelig på unix arbeidsstasjon. Formatene har følgende navn:

SSBw_dV.	Viser tallet med tusenskilte og evt. desimalkomma
SSBw_dX.	Deler tallet på 10 før det vises med tusenskilte og evt. desimalkomma
SSBw_dC.	Deler tallet på 100 før det vises med tusenskilte og evt. desimalkomma
SSBw_dM.	Deler tallet på 1 000 før det vises med tusenskilte og evt. desimalkomma
SSBw_dT.	Deler tallet på 10 000 før det vises med tusenskilte og evt. desimalkomma
SSBw_dH.	Deler tallet på 100 000 før det vises med tusenskilte og evt. desimalkomma
SSBw_dA.	Deler tallet på 1 000 000 før det vises med tusenskilte og evt. desimalkomma

Disse formatene kan brukes på tall med totalt 5-15 siffer (w) hvorav 0-3 desimaler (d). De formaterte tallenes faktiske bredde i tabellen blir større enn totalt antall siffer, fordi desimaltegn og tusenskilte tar plass. Den faktiske bredden kan regnes ut som:

	totalt antall siffer (w)
+	antall tusenskilte
+	eventuelt desimaltegn
+	1

=	faktisk bredde
=====	

Ved å bruke disse formatene sørger vi også for at null skrives ut som bindestrek og at tall som er for store blir skrevet ut som stjerner i tabellen. Hvis dette siste skjer, må vi utvide formatet til å romme større tall og lage tabellen på nytt. Formatet SSB8_1V. gir totalt 8 siffer hvorav 1 desimal og faktisk bredde på 12 tegn.

På grunn av en merkelig endring i TABULATE fra versjon 6.12 til 8.1, vil disse formatene få venstrejustert tekst i utskriftsvinduet for de tabellceller som er tomme, dvs. de som skrives ut som en strek. Dette gjør at tabellen ikke blir som den skal. Velger vi å sende tabellen vår til en html-fil isteden, vil dette problemet forsvinne.

Tabell med flere grupperingsvariable og flere analysevariable

Vi starter med en tabell der vi har flere grupperingsvariable i forspalten og hodet består av flere analysevariable, og der vi både lager summer og prosentuerer i en og samme tabell. Vi ønsker en tabell pr. årgang, og vi ønsker årstallet i overskriften. Da sorterer vi først datasettet etter årgang. Deretter setter vi option NOBYLINE på for å unngå at årstallet kommer på en egen linje i overskriften. I tabellprogrammet bruker vi BY-setningen, for å angi at det skal lages en tabell pr. år. For å plassere årstallet i TITLE-setningen, bruker vi hjelpevariabelen #BYVAL(aar), som inneholder årstallet. Legg merke til at vi plasserer variable *etter hverandre* i forspalten ved ganske enkelt å ramse dem opp, adskilt av mellomrom.



Legg også merke til at vi bruker en ALL*ALL-kombinasjon for totallinjen. Denne, sammen med INDENT=0, gjør at vi får en blank linje før sumlinjen i tabellen. Tilsvarende bruker vi ALL*ALL="Tekst"*ALL før hver grupperingsvariabel. Den første ALL gir oss en blank linje, den andre gir oss tekst for grupperingsvariabelen og den siste gir oss en blank linje før selve grupperingen skrives ut. Denne metoden kan vi bruke så lenge vi bare har et nivå av grupperingsvariable i forspalten. Metoden egner seg ikke når vi skal ha tabellen ut på en html-fil.



For å få tabellen til å se ut nøyaktig slik som vi vil, er det nødvendig med litt 'tricksing'. Det må vi ty til for å gi *flere uavhengige kolonner* en felles overskrift. For eksempel er *innbyggere* og *døde* to helt uavhengige variable (innbyggere og døde). Begge disse, samt kolonnen med innvandrede, skal ha samme overskrift. For å få til dette, setter vi de tre kolonne-uttrykkene inni en felles parentes og plasserer SUM='Antall' utenfor. Dette gir den ønskede effekt.

```
PROC SORT DATA=bef OUT=befs ;
  BY aar;
RUN;

OPTIONS NOBYLINE;
PROC TABULATE DATA=befs FC=' -----' NOSEPS MISSING ORDER=DATA;
  WHERE landsdel IN('1','2','3','4');
  CLASS kjonn landsdel aldersgruppe /PRELOADFMT;
  VAR innbyggere dode innvandrede;
  BY aar;
  TABLE ALL=' '*ALL='I alt'
         ALL=' '*ALL='Landsdel'*ALL=' '*landsdel=' '
         ALL=' '*ALL='Kjønn'*ALL=' '*kjønn=' '
         ALL=' '*ALL='Alder'*ALL=' '*aldersgruppe=' '
         ,
         (SUM='Antall'*F=ssb7_0v. PCTSUM='Prosent'*F=ssb6_1v.)*
         (innbyggere='Innbyg- gere'
         dode='Døde'
         innvandrede='Innvand- rede')
         / RTS=17 MISSTEXT='- ' BOX='Landsdel, kjønn og alder' INDENT=0
  ;
  FORMAT kjonn          $kjonn.
         landsdel       $ldel.
         aldersgruppe   $aldgrp.
  ;
  TITLE 'Tabell 14. Befolkning etter landsdel, kjønn og alder. #BYVAL(aar)';
RUN;
OPTIONS BYLINE;
RUN;
```

Tabell 14. Befolkning etter landsdel, kjønn og alder. 1998						
Landsdel, kjønn og alder	Antall			Prosent		
	Innbyggere	Døde	Innvandrede	Innbyggere	Døde	Innvandrede
I alt	4 417 599	44 112	36 704	100,0	100,0	100,0
Landsdel						
Nord	591 232	5 933	4 047	13,4	13,4	11,0
Vest	1 037 762	10 050	6 505	23,5	22,8	17,7
Øst	2 170 559	22 744	20 740	49,1	51,6	56,5
Sør	618 046	5 385	5 412	14,0	12,2	14,7
Kjønn						
Kvinner	2 232 493	22 045	18 392	50,5	50,0	50,1
Menn	2 185 106	22 067	18 312	49,5	50,0	49,9
Alder						
0-19 år	1 138 732	479	10 685	25,8	1,1	29,1
20-67 år	2 693 777	8 023	25 608	61,0	18,2	69,8
67 år og over	585 090	35 610	411	13,2	80,7	1,1
Tabell 14. Befolkning etter landsdel, kjønn og alder. 1999						
Landsdel, kjønn og alder	Antall			Prosent		
	Innbyggere	Døde	Innvandrede	Innbyggere	Døde	Innvandrede
I alt	4 445 329	45 170	41 841	100,0	100,0	100,0
Landsdel						
Nord	589 605	5 950	5 618	13,3	13,2	13,4
Vest	1 042 923	10 312	7 593	23,5	22,8	18,1
Øst	2 188 257	23 356	22 328	49,2	51,7	53,4
Sør	624 544	5 552	6 302	14,0	12,3	15,1
Kjønn						
Kvinner	2 245 770	22 744	20 977	50,5	50,4	50,1
Menn	2 199 559	22 426	20 864	49,5	49,6	49,9
Alder						
0-19 år	1 147 628	529	13 544	25,8	1,2	32,4
20-67 år	2 712 754	8 044	27 844	61,0	17,8	66,5
67 år og over	584 947	36 597	453	13,2	81,0	1,1

Denne teknikken for å legge inn blanke linjer er ikke brukbar for å lage html-tabeller.

Samme variabel til både gruppering og analyse



Det kan av og til være ønskelig å bruke samme variabel både til gruppering og til analyse. Vi kan f.eks ønske å bruke variabelen *innbyggere* både til å dele inn i grupper samtidig som vi ønsker å summere antall innbyggere. Vi kan ikke oppgi samme variabel både i VAR- og CLASS-setningene, derfor må vi gå veien om en hjelpevariabel. Vi må lage en ny variabel, en hjelpevariabel, som er lik den opprinnelige. Dette ser vi i eksempelet under:

```

DATA bef2;
  SET bef;
  innb2=innbyggere;
RUN;

PROC FORMAT;
  VALUE innb
    1-299999 = 'Under 300 000'
    300000-high = '300 000 og over'
    OTHER = 'Uoppgitt'
  ;
RUN;

PROC TABULATE DATA=bef2 FC=' -----' NOSEPS F=7. MISSING ORDER=DATA;
  WHERE aar='1999' AND landsdel IN('1','2','3','4');
  CLASS kjonn landsdel innb2/PRELOADFMT;
  VAR innbyggere ;
  TABLE (ALL='Hele landet' landsdel=' ')*(ALL='I alt' innb2=' ')
    ,
    (ALL='Begge kjønn' kjonn=' ')*
    SUM=' '*innbyggere=' '
    / RTS=27 MISSTEXT='- ' INDENT=2
  ;
  FORMAT kjonn          $kjonn.
         landsdel       $ldel.
         aldersgruppe   $aldgrp.
         innb2          innb.
  ;
  TITLE 'Tabell 15. Befolkning etter landsdel og kjønn.';
RUN;

```

Tabell 15. Befolkning etter landsdel og kjønn.

	Begge kjønn		
	Kvinner	Menn	
Hele landet			
I alt	4445329	2245770	2199559
Under 300 000	2460642	1262412	1198230
300 000 og over	1984687	983358	1001329
Nord			
I alt	589605	293629	295976
Under 300 000	589605	293629	295976
Vest			
I alt	1042923	523706	519217
Under 300 000	419698	219246	200452
300 000 og over	623225	304460	318765
Øst			
I alt	2188257	1114880	1073377
Under 300 000	826795	435982	390813
300 000 og over	1361462	678898	682564
Sør			
I alt	624544	313555	310989
Under 300 000	624544	313555	310989

Analysevariabel i hodet og statistikk nøkkelord i forspalten

Noen ganger vil det lønne seg å ha statistikkbegrepene i forspaltdimensjonen og analysevariabelen som hodedimensjon. Eksempelet under viser en tabell vi lager på grunnlag av en AKU-fil (for beskrivelse av filen, se vedlegg, side 76). Dette er en utvalgsfil som har en observasjon pr. person i utvalget. For å blåse opp tallene i utvalgsfiler til å gjelde hele populasjonen bruker vi vekter. Vi skal summere vektene for 1999 fordelt etter landsdel i forspalten og om personen er i arbeid i hodet. Her har vi dessuten bestemt oss for å ta med antall i utvalget summeringene er basert på som siste linje i tabellen. For å få til denne tabellen, må SUM og N defineres i forspaltdimensjonen. Analysevariabelen *vekt1999* vil det være enklest å definere i hodedimensjonen. Vi må da være klar over at N vil omfatte de observasjoner der *vekt1999* ikke er missing.

For å skille mellom de forskjellige delene av forspalten, har vi trickset litt. Vi har brukt samme metode som i eksempelet som lager tabell 14 for å få inn luft i tabellen. Dette gjør at tabellen blir lettere å lese.



```

PROC TABULATE DATA=aku NOSEPS MISSING FC='-----' F=10.;
  CLASS landsdel iarbeid1999;
  VAR vekt1999;
  TABLE (ALL='I alt' ALL=' '*landsdel=' ')*
    SUM=' ' ALL=' '*N='Antall i utvalget',
    (ALL='I alt' iarbeid1999='I arbeid?')*vekt1999=' '
    /RTS=25 MISSTEXT='- ' BOX='Landsdel' INDENT=0;
  FORMAT landsdel $ldel.
    iarbeid1999 $svar.
  ;
  TITLE 'Tabell 16. Personer med og uten arbeid. 1999';
RUN;
    
```

Tabell 16. Personer med og uten arbeid. 1999

Landsdel	I arbeid?			
	I alt	Uoppgitt	Ja	Nei
I alt	2122427	159387	1673865	289175
Øst	900336	48982	664814	186540
Sør	382402	-	336767	45635
Vest	531497	110405	421092	-
Nord	308192	-	251192	57000
Antall i utvalget	43	3	34	6

Veiing, oppblåsing

I utvalgsundersøkelser må hver observasjon multipliseres med en faktor (veies med en vekt, blåses opp med en oppblåsningsfaktor) for at datasettet skal kunne representere hele populasjonen. Denne faktorens (vekten) verdi vil kunne variere fra observasjon til observasjon. Vekten må ligge som en egen variabel på SAS datasettet. For at hver observasjon skal veies med vekten, bruker vi vanligvis WEIGHT-setningen i TABULATE. Vekten kan gjerne være et desimaltall (i motsetning til variable brukt i FREQ-setningen, som kun benytter heltall).

Vi må være klar over at statistikkene N, NMISS og PCTN ikke veies! Dette fordi vi normalt ikke veier N, men tvertimot benytter N for å vise hvor mange observasjoner beregningene faktisk baserer seg på.

Dersom vi likevel ønsker å veie N, må vi bruke statistikk nøkkelordet SUMWGT som *må* knyttes til en analysevariabel. Bemerk at SUMWGT ignorerer vektene for observasjoner der analysevariabelen mangler verdi!

Neste eksempel viser to tabeller fra en utvalgsundersøkelse. Den ene viser uveide tall, den andre er med veide tall.

I SAS-datasettet som brukes i eksempelet under, representerer hver observasjon en person som er med i utvalget. For at datasettet skal representere *alle* i hele populasjonen, må hver observasjon med tilhørende vekt summeres. Vi vil beregne forskjellige statistikker for variabelen *arbtid1999* og denne skal i den veide tabellen benytte *vekt1999* til dette. Vi ønsker dessuten å vite hvor mange observasjoner beregningene er basert på (uveiet N), samt totalt antall personer (veiet, oppblåst N).

```

PROC TABULATE DATA=aku FC='-----' NOSEPS MISSING F=8.;
  CLASS landsdel;
  VAR arbtid1999;
  WEIGHT vekt1999;
  TABLE ALL='I alt' landsdel=' '
        ,
        arbtid1999=' '*
        (MEAN='Gj.snitt'*F=numx9.1 SUM STD='Standardavvik'*F=numx9.2
        SUMWGT='Antall' N)
  /RTS=10 MISSTEXT='- ' BOX='Landsdel' ;
  FORMAT landsdel $ldel.
  ;
  TITLE 'Tabell 17. Arbeidstid 1999. Veide tall' ;
RUN;

PROC TABULATE DATA=aku FC='-----' NOSEPS MISSING F=8.;
  CLASS landsdel;
  VAR arbtid1999;
  TABLE ALL='I alt' landsdel=' '
        ,
        arbtid1999=' '*
        (MEAN='Gj.snitt'*F=numx9.1 SUM STD='Standardavvik'*F=numx9.2 N)
  /RTS=10 MISSTEXT='- ' BOX='Landsdel';
  FORMAT landsdel $ldel.
         kjonn      $kjonn.
  ;
  TITLE 'Tabell 18. Arbeidstid 1999. Uveide tall' ;
RUN;

```

Tabell 17. Arbeidstid 1999. Veide tall

Landsdel	Gj.snitt	Sum	Standard- avvik	Antall	N
I alt	40,0	69689542	1283,42	1741170	35
Øst	39,2	27966676	1156,08	713796	15
Sør	40,7	13700227	1011,12	336767	7
Vest	41,9	18414753	1860,11	439415	8
Nord	38,2	9607887	991,73	251192	5

Tabell 18. Arbeidstid 1999. Uveide tall

Landsdel	Gj.snitt	Sum	Standard- avvik	N
I alt	40,1	1443	5,80	36
Øst	39,1	626	5,12	16
Sør	40,9	287	4,72	7
Vest	42,5	340	8,35	8
Nord	38,0	190	4,27	5

Legg merke til at gjennomsnittstallene bare endrer seg litt fra de veide til de uveide tallene. Legg også merke til at summen i N-kolonnen er større i den uveide tabellen enn i den veide. Dette viser at det finnes en observasjon som ikke har vekt, hvilket helst ikke skal forekomme. Når slike uovenstemmelser oppdages, bør vi sjekke datafilen vår nøye.

I eksempelet over benytter vi bare én vektvariabel. Vi kan bruke forskjellige vektvariable til forskjellige variable. For å få til det må vi bruke WEIGHT-opsjonen i VAR-setningen:

```

PROC TABULATE DATA=aku FC=' -----' NOSEPS MISSING F=8.;
  CLASS landsdel;
  VAR arbtid1996/WEIGHT=vekt1996;
  VAR arbtid1997/WEIGHT=vekt1997;
  VAR arbtid1998/WEIGHT=vekt1998;
  VAR arbtid1999/WEIGHT=vekt1999;
  TABLE ALL='I alt' landsdel=' '
    ,
    (
      arbtid1996='1996'
      arbtid1997='1997'
      arbtid1998='1998'
      arbtid1999='1999'
    ) *MEAN=' '*F=numx9.1
  /RTS=10 MISSTEXT='- ' BOX='Landsdel' ;
  FORMAT landsdel $ldel.
  ;
  TITLE ' Tabell 19. Gjennomsnittlig arbeidstid 1996-1999. Veide tall' ;
RUN;

```

Tabell 19. Gjennomsnittlig arbeidstid 1996-1999. Veide tall

Landsdel	1996	1997	1998	1999
I alt	37,0	37,2	38,5	40,0
Øst	36,0	37,5	38,6	39,2
Sør	37,5	39,3	37,8	40,7
Vest	38,4	35,7	38,7	41,9
Nord	35,8	36,8	38,6	38,2

En tredje variant har vi når vi f.eks. ønsker å summere antall personer (veid antall) fordelt etter forskjellige grupperingsvariable. Vi skal da egentlig ikke trenge noen analysevariabel, vi ønsker bare å få summert vektene. Det er bare en liten hake ved det og det er at vi må ha en analysevariabel for å få lov til å bruke vekter. Når vi nå ikke har bruk for noen analysevariabel, men kun ønsker å summere vektene, bruker vi vekten som en analysevariabel, slik som i eksempelet under.

```

PROC TABULATE DATA=aku FC=' -----' NOSEPS MISSING F=8.;
  CLASS landsdel iarbeid1999;
  VAR vekt1999;
  TABLE ALL='I alt' landsdel=' '
        '
        (ALL='I alt' iarbeid1999='I arbeid?')
        *vekt1999=' '*SUM=' '
  /RTS=10 MISSTEXT='- ' BOX='Landsdel' ;
  FORMAT landsdel $ldel.
         iarbeid1999 $svar.
        ;
  TITLE ' Tabell 20. Personer med og uten arbeid 1999. Veide tall' ;
RUN;

```

Tabell 20. Personer med og uten arbeid 1999. Veide tall

Landsdel	I arbeid?			
	I alt	Uoppgitt	Ja	Nei
I alt	2122427	159387	1673865	289175
Øst	900336	48982	664814	186540
Sør	382402	-	336767	45635
Vest	531497	110405	421092	-
Nord	308192	-	251192	57000

Antall i alt i denne tabellen og i det første eksempelet i dette kapittelet (tabell 17) er forskjellige. Det skyldes at det finnes MISSING-verdier på flere forskjellige variable i SAS datasettet. I eksempelet over summeres alle vektene, mens i tabell 17 ble vektene summert for de observasjonene som har verdi for variabelen *arbtid1999*. Dette viser at det er viktig å ha oversikt over MISSING-verdier i SAS datasett og være klar over hvilken effekt de har på tabelleringen vår.

Gjøre om en grupperingsvariabel til en analysevariabel

I intervjuundersøkelser stilles det ofte ved spørsmål der intervjuobjektet skal svare enten ja eller nei. Disse svarene kodes oftest med 1 for ja og 2 for nei. I tillegg til disse to svarene kan intervjuobjektet nekte å svare eller han vet ikke hva han skal svare. Disse svarene gis også egne koder i datafilene. Variablene som inneholder svarene på ja/nei-spørsmål er normalt definert som grupperingsvariable. Når vi lager en tabell med disse, lages det en kolonne eller rad for hvert svaralternativ. I tabellen under spørres det om personen er i arbeid eller ei, og. Det er en kolonne for hvert svaralternativ. (Alle som ikke har svart enten ja eller nei er kodet til blank.)

```

PROC FORMAT;
VALUE $svar
  '1'   = 'Ja'
  '2'   = 'Nei'
  OTHER = 'Uoppgitt'
;
PROC TABULATE DATA=aku NOSEPS MISSING FC='-----' F=10.;
CLASS landsdel iarbeid1999;
VAR vekt1999;
TABLE all='I alt' landsdel=' ' ,
      (ALL='I alt' iarbeid1999='I arbeid?')*
      ROWPCTSUM=' '*vekt1999=' ' vekt1999=' '*N*F=5.
      /RTS=15 MISSTEXT='-' BOX='Landsdel';
FORMAT landsdel $ldel.
      iarbeid1999 $svar.
;
TITLE 'Tabell 21. Personer med og uten arbeid. Prosent. 1999';
RUN;
    
```

Her er tabellen programmet ovenfor lager:

Tabell 21. Personer med og uten arbeid. Prosent. 1999

Landsdel	I arbeid?				N
	I alt	Uoppgitt	Ja	Nei	
I alt	100	8	79	14	43
Øst	100	5	74	21	19
Sør	100	-	88	12	8
Vest	100	21	79	-	10
Nord	100	-	82	18	6



Tabellen er grei nok, men ofte er vi bare interessert i kolonnen med ja-svarene. For å kunne lage en tabell som bare inneholder den prosent som har svart ja på spørsmålet, må vi først bearbeide dataene våre litt. Vi må gjøre om grupperingsvariabelen for svaret til en analysevariabel som har verdien 1 når personen har svart ja og MISSING når han har svart nei. Denne nye hjelpevariabelen skal vi bruke som teller i prosentberegningen vår. Vi trenger også en nevner i prosentueringen, den skal ha verdien 1 for de som har svart enten ja eller nei på spørsmålet. De som ikke har svart, ikke vet eller nekter å svare holder vi utenfor. Dette fordi vi ikke kan vite om de ville ha svart ja eller nei. Tar vi dem med antar vi samtidig at de ville ha svart nei og ja-prosenten kan bli lavere enn den skal være. Om de som ikke har svart på spørsmålet skal være med i nevneren, må vi avklare før vi lager tabellen. Hvis denne gruppen er stor, gir det stor forskjell på ja-andelen om vi har dem med eller ikke. I dette eksempelet holder vi dem altså utenfor, vi lager en teller og en nevner for hver variabel vi skal prosentuerer. Tellervariabelen (i eksemplet kalt *ia1999*) får altså verdien 1 når personen har svart ja og nevnervariabelen (*svaria1999*) får verdi 1 når personen har svart enten ja eller nei på spørsmålet.

N-kolonnen i tabellen over er knyttet til analysevariabelen *vekt1999* fordi det er bare de som har en verdi for *vekt1999* som kommer med i resten av tabellen og følgelig er det de som utgjør N. Dermed blir det overensstemmelse mellom N-kolonnen og prosentkolonnene for antallet observasjoner beregningene er utført på.

```
* Lager hjelpevariable for prosenttabell;
DATA aku2;
SET aku ;
IF iarbeid1999 = '1' THEN
    ia1999 = 1;
IF iarbeid1999 IN('1','2') THEN
    svaria1999 = 1;
LABEL ia1999      = 'I arbeid i perioden (=1)'
      svaria1999 = 'Har svart på spørsmål om arbeid'
      ;
RUN;
```

I tabellprogrammet bruker vi hjelpevariablene vi laget i data-steget som analysevariable. Vi bruker PCTSUM for prosentueringen. De som har svart ja telles opp og regnes i prosent av de som har svart ja eller nei.

```
PROC TABULATE DATA=aku2 NOSEPS MISSING FC='-----' F=10.;
CLASS landsdel ;
VAR ia1999 svaria1999;
WEIGHT vekt1999;
TABLE all='I alt' landsdel=' ' ,
      ia1999='I arbeid'*PCTSUM<svaria1999>=' ' svaria1999=' '*N*F=5.
      /RTS=15 MISSTEXT='- ' BOX='Landsdel';
FORMAT landsdel $ldel.
      ;
TITLE 'Tabell 22. Personer i arbeidsstyrken i arbeid. Prosent. 1999';
RUN;
```

Av tabellen ser vi at prosenttallene for de som har svart ja er større enn de var i den forrige tabellen. Det er fordi de som ikke har svart ja eller nei på spørsmålet holdes utenfor prosentueringen. Vi har her tatt med en N-kolonne for å vise hvor mange observasjoner prosentueringen er basert på. Legg merke til at denne er knyttet mot analysevariabelen *svaria1999*. Dette gjøres fordi vi ønsker å telle opp kun de observasjoner som har en verdi for *svaria1999*.

Når vi veier tallene våre med en vekt ganges hvert tall som skal summeres med vekten før summeringen utføres. Mangler vekten verdi (er missing) vil det ikke bli beregnet noen veid verdi. Observasjoner med manglende vekt vil heller ikke komme med i antallet observasjoner som har en verdi for en analysevariabel (N). For at N skal telles opp, og den veies aldri, må både variabelen N er knyttet til og vekten ha verdi. I tabellen under viser N-kolonnen hvor mange observasjoner det er som har verdi for variabelen *svaria1999* og samtidig en verdi for vekten *vekt1999*.

Tabell 22. Personer i arbeidsstyrken i arbeid. Prosent. 1999

Landsdel	I arbeid	N
I alt	85	40
Øst	78	18
Sør	88	8
Vest	100	8
Nord	82	6

Av og til ønsker vi å få ut andelstall istedenfor prosenttall. Når vi skal det kan vi bruke samme prinsipp som over, men vi teller de som har svart ja med 0,01 istedenfor 1. Da vil vi få andelstall når vi prosentuerer i tabellprogrammet.

```

DATA aku2;
SET aku ;
IF iarbeid1999 = '1' THEN
  ia1999 = 0.01;
ELSE
  ia1999 = .;
IF iarbeid1999 IN('1','2') THEN
  svaria1999 = 1;

LABEL ia1999      = 'I arbeid i perioden (=1)'
      svaria1999 = 'Har svart på spørsmål om arbeid'
      ;
RUN;

PROC TABULATE DATA=aku2 NOSEPS MISSING FC=' -----' F=10.;
  CLASS landsdel ;
  VAR ia1999 svaria1999;
  WEIGHT vekt1999;
  TABLE all='I alt' landsdel=' ' ,
         ia1999='Andel i arbeid'*PCTSUM<svaria1999>=' '*F=numx7.2
         svaria1999=' '*N*F=5.
         /RTS=15 MISSTEXT='- ' BOX='Landsdel';
  FORMAT landsdel  $ldel.
      ;
  TITLE 'Tabell 23. Personer i arbeidsstyrken i arbeid. 1999';
RUN;

```

Nå får vi andelstall i tabellen vår.

Tabell 23. Personer i arbeidsstyrken i arbeid. 1999

Landsdel	Andel i arbeid	N
I alt	0,85	40
Øst	0,78	18
Sør	0,88	8
Vest	1,00	8
Nord	0,82	6

Ratetabeller

Å lage ratetabeller med TABULATE krever noe mer programmering i forkant av tabelleringen enn de tabellene vi hittil har sett på. Vi skal her se på et eksempel der vi lager en tabell med rater pr. 1000 innbyggere for AKU-tallene våre. I tillegg til AKU-dataene trenger vi befolkningsdata. La oss først se hvordan tabellen skal se ut:

Landsdel og kjønn	1996	1997	1998	1999	Personer i arbeids- styrken pr. 1000 innbyggere 1999
	I alt				
Begge kjønn	1942312	1988254	1972675	2037463	458
Menn	1052451	1078502	1078784	1133778	515
Kvinner	889861	909752	893891	903685	402
Øst					
Begge kjønn	827237	846668	839949	851354	389
Menn	459264	469276	471558	467792	436
Kvinner	367973	377392	368391	383562	344
Sør					
Begge kjønn	334050	342918	339271	382402	612
Menn	223000	229472	226567	272202	875
Kvinner	111050	113446	112704	110200	351
Vest					
Begge kjønn	489280	501212	500223	495515	475
Menn	191744	196656	199452	200692	387
Kvinner	297536	304556	300771	294823	563
Nord					
Begge kjønn	291745	297456	293232	308192	523
Menn	178443	183098	181207	193092	652
Kvinner	113302	114358	112025	115100	392

Prosessen for å lage denne tabellen er slik:

- Lag et befolkningsdatasett med innbyggere fordelt etter landsdel og kjønn.
- Aggregér datafilen til landsdel*kjønn
- Koble AKU-data med befolkningsdata
- Lag tabellen

Vi sørger for å aggregere befolkningsdataene til riktig nivå for tabellering. Samtidig sørger vi for å fjerne data med uoppgitt kjønn.

```
PROC MEANS DATA=bef NOPRINT NWAY;
WHERE aar='1999' AND kjønn IN('1','2');
CLASS landsdel kjønn;
VAR innbyggere;
OUTPUT OUT=befaggregert (DROP=_TYPE_ _FREQ_) SUM= ;
RUN;
```

AKU-dataene inneholder en utvalgt person pr. observasjon. De aggregeres til landsdel * kjønn, slik som vi tidligere har gjort med befolkningsdataene.

```
PROC MEANS DATA=aku NOPRINT NWAY;
WHERE kjønn IN('1','2') ;
CLASS landsdel kjønn ;
VAR vekt1996-vekt1999;
OUTPUT OUT=akuaggregert (DROP=_TYPE_ _FREQ_) SUM= ;
RUN;
```

Datasettene skal nå kobles etter landsdel og og kjønn. Fordi vi skal beregne tall pr. 1000 innbyggere istedenfor pr. 100 (som tilsvarer prosent), deler vi innbyggertallene på 10.:

```
DATA akumedbef ;
  MERGE akuaggregert (IN=ds1)
        befaggregert (IN=ds2)
        ;
  BY landsdel kjonn ;
  innbyggere=innbyggere/10;
  IF ds1 ;
RUN;
```

Nå kan vi lage tabellen og bruke prosentuering for å lage ratetallene:

```
PROC TABULATE DATA=akumedbef NOSEPS MISSING FC=' -----' F=10.;
  CLASS landsdel kjonn ;
  VAR vekt1996-vekt1999 innbyggere;
  TABLE (all='I alt' landsdel=' ')*(ALL='Begge kjønn' kjonn=' '),
    vekt1996='1996'*SUM=' '
    vekt1997='1997'*SUM=' '
    vekt1998='1998'*SUM=' '
    vekt1999='1999'*SUM=' '
    vekt1999='Personer i arbeids- styrken pr. 1000 innbyggere 1999'*
    PCTSUM<innbyggere>=' '
    /RTS=15 MISSTEXT='- ' BOX='Landsdel og kjønn' INDENT=2 PRINTMISS;
  FORMAT landsdel $ldel.
         kjonn $kjonn.
         ;
  TITLE 'Tabell 24. AKU rater';
RUN;
```

Ikke-hierarkiske tabeller

Med TABULATE lager vi helst hierarkiske tabeller, dvs. tabeller der alle undernivåer summerer seg til overliggende nivå. Det er slike tabeller TABULATE først og fremst er laget for. Av og til hender det dog at vi ønsker å lage tabeller der undernivående ikke alltid summerer seg til overliggende nivå, eller der det for noen overliggende nivåer ikke skal spesifiseres noe undernivå. Tabellen under viser et eksempel på en slik tabell.

Tabell 25. Personer i arbeidsstyrken etter næring.

		1996	1997	1998	1999
01-45	I ALT	2020778	2067524	2051317	2122427
01	JORDBRUK, JAKT OG VILTSTELL	45436	46664	47363	48323
02	SKOGBRUK	33331	34346	34241	38811
05	FISKE, FANGST OG FISKEOPPDRETT	99765	102147	98592	103735
11	UTV.AV RÅOLJE OG NATURGASS, INKL.TJ.	33331	34346	34241	38811
10,12-14	BERGVERKSDRIFT	225164	231061	232682	228880
15-37	INDUSTRI	1416422	1448832	1436044	1484339
15-16	Nærings- og nytelsesmiddelindustri	165112	169752	164966	168381
17-19	Tekstil- og bekledningsindustri	112098	115356	115845	135945
20	Trelast- og trevareindustri	55525	56723	56352	53100
21	Treforedling	44240	45424	42240	45635
22	Forlag og grafisk industri	45436	46664	47363	49323
23-25	Oljeraffinering, kjemisk og plast	245164	251061	252682	255280
26	Mineralproduktindustri	33331	34346	34241	48811
27	Metallindustri	113302	114358	112025	110100
28	Metallvareindustri	45436	46664	47363	-
29	Maskinindustri	33331	34346	34241	38811
30-33	Elektroteknisk og optisk industri	355004	361040	357519	382861
34-35	Transportmiddelindustri	89676	92088	89603	97958
36-37	Møbelindustri og annen industri	78767	81010	81604	98134
40-41	KRAFT- OG VANNFORSYNING	88856	91069	90593	94911
45	BYGGE- OG ANLEGGSVIRKSOMHET	78473	79059	77561	84617

Denne tabellen har to nivåer av næring i forspalten, disse er skilt ved at det øverste nivået har store bokstaver i forspalteteksten, mens det underliggende har små bokstaver. Med vanlig bruk av TABULATE vil vi kunne få en slik hierarkisk tabell:

Tabell 25b. Personer i arbeidsstyrken etter næring.

		1996	1997	1998	1999
01-45	I ALT	2020778	2067524	2051317	2122427
01	JORDBRUK, JAKT OG VILTSTELL				
		45436	46664	47363	48323
01	Jordbruk, jakt og viltstell	45436	46664	47363	48323
02	SKOGBRUK				
		33331	34346	34241	38811
02	Skogbruk	33331	34346	34241	38811
05	FISKE, FANGST OG FISKEOPPDRETT				
		99765	102147	98592	103735
05	Fiske, fangst og fiskeoppdrett	99765	102147	98592	103735
11	UTV.AV RÅOLJE OG NATURGASS, INKL.TJ.				
		33331	34346	34241	38811
11	Utv.av råolje og naturgass,inkl.tj.	33331	34346	34241	38811
10,12-14	BERGVERKSDRIFT				
		225164	231061	232682	228880
10,12-14	Bergverksdrift	225164	231061	232682	228880
15-37	INDUSTRI				
		1416422	1448832	1436044	1484339
15-16	Nærings- og nytelsesmiddelindustri	165112	169752	164966	168381
17-19	Tekstil- og bekledningsindustri	112098	115356	115845	135945
20	Trelast- og trevareindustri	55525	56723	56352	53100
21	Treforedling	44240	45424	42240	45635
22	Forlag og grafisk industri	45436	46664	47363	49323
23-25	Oljeraffinering, kjemisk og plast	245164	251061	252682	255280
26	Mineralproduktindustri	33331	34346	34241	48811
27	Metallindustri	113302	114358	112025	110100
28	Metallvareindustri	45436	46664	47363	-
29	Maskinindustri	33331	34346	34241	38811
30-33	Elektroteknisk og optisk industri	355004	361040	357519	382861
34-35	Transportmiddelindustri	89676	92088	89603	97958
36-37	Møbelindustri og annen industri	78767	81010	81604	98134
40-41	KRAFT- OG VANNFORSYNING				
		88856	91069	90593	94911
40-41	Kraft- og vannforsyning	88856	91069	90593	94911
45	BYGGE- OG ANLEGGSVIRKSOMHET				
		78473	79059	77561	84617
45	Bygge- og anleggsvirksomhet	78473	79059	77561	84617

Denne tabellen er ikke så pen som ønskelig, men den er hierarkisk oppbygd ettersom alle undernivåer summerer seg opp. Slik denne forspalten er bygd opp er det noen næringer som bare har et undernivå, f.eks. skogbruk, mens andre har flere (industri). De næringene som bare har et undernivå ønsker vi ikke å ta med i tabellen. Vi vil da bryte hierarkiet i tabellen.

Ved å definere såkalte MULTILABEL-formater med PROC FORMAT kan vi lage ikke-hierarkiske tabeller med TABULATE. TABULATE er en av de få prosedyrene i SAS som takler disse MULTILABEL-formatene. I tillegg til å definere et format som MULTILABEL, må vi sørge for å bestemme rekkefølgen formatet skal lagres i for senere å kunne bruke denne rekkefølgen. Det gjør vi med option NOTSORTED.

```

PROC FORMAT;
  VALUE $naring (NOTSORTED MULTILABEL)
    '01'-'45'      = '01-45      I ALT'
    '01'           = '01         JORDBRUK, JAKT OG VILTSTELL'
    '02'           = '02         SKOGBRUK'
    '05'           = '05         FISKE, FANGST OG FISKEOPPDRETT'
    '11'           = '11         UTV.AV RÅOLJE OG NATURGASS, INKL.TJ.'
    '10','12'-'14' = '10,12-14 BERGVERKSDRIFT'
    '15'-'37'      = '15-37      INDUSTRI'
    '15'-'16'      = '15-16      Nærings- og nytelsesmiddelindustri'
    '17'-'19'      = '17-19      Tekstil- og bekledningsindustri'
    '20'           = '20         Trelast- og trevareindustri'
    '21'           = '21         Treforedling'
    '22'           = '22         Forlag og grafisk industri'
    '23'-'25'      = '23-25      Oljeraffinering, kjemisk og plast'
    '26'           = '26         Mineralproduktindustri'
    '27'           = '27         Metallindustri'
    '28'           = '28         Metallvareindustri'
    '29'           = '29         Maskinindustri'
    '30'-'33'      = '30-33      Elektroteknisk og optisk industri'
    '34'-'35'      = '34-35      Transportmiddelindustri'
    '36'-'37'      = '36-37      Møbelindustri og annen industri'
    '40'-'41'      = '40-41      KRAFT- OG VANNFORSYNING'
    '45'           = '45         BYGGE- OG ANLEGGSVIRKSOMHET'
  ;
RUN;

```

Vi ser at samme næringskode gjentas i flere linjer. Vi har også passet på å lage den rekkefølgen på tekstlinjene som vi ønsker å få ut i tabellen vår. Vi har også tatt med totalinjen øverst ved å la den inneholde alle aktuelle næringskoder. Her er det viktig å passe på at det ikke er næringskoder på SAS-datasettet som ikke er med i noen enkeltlinjer, men som er inkludert i totalen. I eksempelet over må vi for eksempel holde næringskode 03 og 04 utenfor SAS-datasettet vårt (eller selektere dem vekk når tabellen lages), hvis de ikke skal være med i totalen. Hvis ikke vil de næringene komme med i total-summen, men aldri være representert i noen enkeltlinje i tabellen.

Selve tabellprogrammet ser slik ut:

```

PROC TABULATE DATA=aku FC=' -----' NOSEPS MISSING F=7. order=data ;
  CLASS naering /preloadfmt mlf ;
  VAR vekt1996-vekt1999;
  TABLE naering=' '
    (
      vekt1996='1996'
      vekt1997='1997'
      vekt1998='1998'
      vekt1999='1999'
    ) *SUM=' '
  /RTS=47 MISSTEXT='- ' ;
  FORMAT naering $naring.;
  TITLE 'Tabell 25. Personer i arbeidsstyrken etter næring.' ;
RUN;

```


TABULATE kontra andre PROC'er

TABULATE istedenfor PROC MEANS

Vi kan bruke TABULATE til å lage samme type deskriptiv statistikk på numeriske variable som den vi får med PROC MEANS. Ved å bruke TABULATE istedenfor PROC MEANS, vil vi kunne bestemme mer over tabellens utseende. Programmet blir ikke vesentlig større eller mer komplisert, men utseende blir altså langt bedre. Blant annet kan vi da bestemme antall desimaler for hvert statistikk nøkkelord. Et eksempel:

```

PROC TABULATE DATA=aku NOSEPS MISSING FC=' -----' F=10.;
  VAR vekt1996-vekt1999 arbtid1996-arbtid1999;
  TABLE vekt1996 vekt1997 vekt1998 vekt1999
         arbtid1996 arbtid1997 arbtid1998 arbtid1999
         ,
         (N*F=5. SUM MEAN*F=numx9.2 MIN*F=5. MAX STD*F=numx9.1)
         /RTS=20 MISSTEXT='- ' ;
  TITLE 'Tabell 26. Deskriptiv statistikk på numeriske variable' ;
RUN;

```

Tabell 26. Deskriptiv statistikk på numeriske variable

	N	Sum	Mean	Min	Max	Std
Vekt 1996	44	2020778	45926,77	33331	75436	10062,8
Vekt 1997	44	2067524	46989,18	33635	77664	10104,1
Vekt 1998	44	2051317	46620,84	34241	75363	10141,5
Vekt 1999	43	2122427	49358,77	35982	74423	7851,7
Arbeidstid 1996	29	1076	37,10	24	63	6,4
Arbeidstid 1997	27	1006	37,26	24	43	3,1
Arbeidstid 1998	34	1315	38,66	24	57	4,8
Arbeidstid 1999	36	1443	40,07	33	60	5,8

Med TABULATE kan vi også lage deskriptiv statistikk etter grupperingsvariable. Her har TABULATE mye større muligheter enn PROC MEANS. Blant annet er det lett å få ut totaler i tillegg til grupperingene. Videre kan vi ha flere forskjellige variable etter hverandre i forspalten, noe eksempelet under viser.

```

PROC TABULATE DATA=aku NOSEPS MISSING FC=' -----' F=10.;
  CLASS landsdel kjonn iarbeid1999;
  VAR vekt1999 ;
  TABLE ALL='I alt'
    landsdel='LANDSDEL'
    kjonn='KJØNN'
    iarbeid1999='I ARBEID?'
    ,
    vekt1999=' '*
    (N*F=5. SUM MEAN*F=numx9.2 MIN*F=5. MAX STD*F=numx9.1)
  /RTS=20 MISSTEXT='- '
  ;
  FORMAT landsdel $ldel.
    kjonn $kjonn.
    iarbeid1999 $svar.
  ;
  TITLE 'Tabell 27. Deskriptiv statistikk vekt1999 etter' ;
  TITLE2 ' landsdel, kjønn og om personene er i arbeid' ;
RUN;

```

Tabell 27. Deskriptiv statistikk vekt1999 etter landsdel, kjønn og om personene er i arbeid

	N	Sum	Mean	Min	Max	Std
I alt	43	2122427	49358,77	35982	74423	7851,7
LANDSDEL						
Øst	19	900336	47386,11	38811	67133	7139,2
Sør	8	382402	47800,25	38811	57100	5917,1
Vest	10	531497	53149,70	35982	74423	9970,0
Nord	6	308192	51365,33	38811	58100	7361,0
KJØNN						
Uoppgitt	2	84964	42482,00	35982	48982	9192,4
Menn	24	1133778	47240,75	38811	67133	6877,5
Kvinner	17	903685	53157,94	38811	74423	7783,2
I ARBEID?						
Uoppgitt	3	159387	53129,00	35982	74423	19553,2
Ja	34	1673865	49231,32	38811	67133	7108,5
Nei	6	289175	48195,83	45635	57000	4600,3

TABULATE istedenfor PROC FREQ

TABULATE kan også brukes til å lage frekvenstabeller, istedenfor PROC FREQ. Ser vi bort fra kumulerte frekvens- og prosenttall, vil vi kunne lage enkle oversiktlige frekvenstabeller med TABULATE. Vi kan lage flere forskjellige frekvenstabeller med TABULATE enn vi kan med PROC FREQ. F.eks kan vi ha flere variable både i forspalte og hode. Eksempelet under viser hvordan vi kan lage en frekvenstabell med TABULATE.

```

PROC TABULATE DATA=aku NOSEPS MISSING FC='-----' F=10.;
  CLASS landsdel kjonn ;
  VAR vekt1999 ;
  TABLE (N='Antall'
          PCTN='% av total'
          ROWPCTN='% av linje'
          COLPCTN='% av kolonne' ) *
          (ALL='I alt' landsdel=''),
          (ALL='I alt' kjonn='')
  /MISSTEXT='- ' RTS=15 INDENT=2;
;
  FORMAT landsdel $ldel.
         kjonn $kjonn.
;
  TITLE 'Tabell 28. Frekvenstabell for landsdel og kjøn'n' ;
RUN;

```

Tabell 28. Frekvenstabell for landsdel og kjøn'n

	I alt	Uoppgitt	Menn	Kvinner

Antall				
I alt	44	2	25	17
Øst	20	1	11	8
Sør	8	-	6	2
Vest	10	1	4	5
Nord	6	-	4	2
% av total				
I alt	100	5	57	39
Øst	45	2	25	18
Sør	18	-	14	5
Vest	23	2	9	11
Nord	14	-	9	5
% av linje				
I alt	100	5	57	39
Øst	100	5	55	40
Sør	100	-	75	25
Vest	100	10	40	50
Nord	100	-	67	33
% av kolonne				
I alt	100	100	100	100
Øst	45	50	44	47
Sør	18	-	24	12
Vest	23	50	16	29
Nord	14	-	16	12

Manglende verdier, missing

Det er viktig å vite at manglende verdier (missing) ikke er det samme som null. Null tas med i statistiske beregninger, missing tas *ikke* med. Innholdet i tabellen vil dermed være avhengig av om verdier er null eller missing.

Manglende verdier i grupperingsvariable



TABULATE *utelater* observasjoner som mangler verdi i en eller flere av *grupperingsvariablene*, hvis vi ikke spesielt ber om at de tas med i opptellingen. **Dette gjelder selv om vi ikke bruker grupperingsvariablen i TABLE-setningen!** Derfor er det viktig at vi i CLASS-setningen bare har med variable vi virkelig skal bruke i tabellen.

Hvis vi vil at TABULATE skal inkludere manglende verdier, må vi ha med MISSING som valg i TABULATE-setningen.

Her følger noen eksempler på hva som skjer hvis vi har missing i grupperingsvariable:

```
DATA test;
  INPUT sivstat $ 1 1  kjonn $ 3 3   fylke $ 5 6   ;
  LABEL sivstat='Sivil status' kjonn='Kjønn'  fylke='Fylke' ;
  CARDS;
1 1 01
2 1
3 1
3 2 03
  ;

PROC TABULATE DATA=test FC='          ' NOSEPS F=8.;
  CLASS sivstat kjonn ;
  TABLE ALL='I alt' sivstat,
         (ALL='I alt' kjonn)*N='Antall'/MISSTEXT=' ';
  TITLE 'Riktig, FYLKE er hverken med i CLASS eller TABLE.';
RUN;
```

Riktig, FYLKE er hverken med i CLASS eller TABLE.

	Kjønn		
	I alt	1	2
	Antall	Antall	Antall
I alt	4	3	1
Sivil status			
1	1	1	-
2	1	1	-
3	2	1	1



Å ta med en variabel i CLASS-setningen som *ikke* skal være med i tabellen, kan få skjebnesvangre følger: Hvis denne variabelen har missing verdier, blir tabellen helt feil:

```
PROC TABULATE DATA=test FC='          ' NOSEPS F=8.;
  CLASS sivstat kjonn fylke ;
  TABLE ALL='I alt' sivstat,
         (ALL='I alt' kjonn)*N='Antall'/MISSTEXT=' ';
  TITLE 'Feil, FYLKE er med i CLASS, men ikke i TABLE.';
RUN;
```

Feil, FYLKE er med i CLASS, men ikke i TABLE.

	Kjønn		
	I alt	1	2
	Antall	Antall	Antall
I alt	2	1	1
Sivil status			
1	1	1	-
3	1	-	1



Ved å ta med MISSING-option i TABULATE-setningen, kommer observasjoner med manglende verdier i grupperingsvariable likevel med, og tabellen blir riktig igjen :

```
PROC TABULATE DATA=test FC='          ' NOSEPS F=8. MISSING ;
  CLASS sivstat kjonn fylke ;
  TABLE ALL='I alt' sivstat,
         (ALL='I alt' kjonn)*N='Antall'/MISSTEXT=' ';
  TITLE 'OK. FYLKE er med i CLASS, men ikke i TABLE. Med MISSING option';
RUN;
```

OK. FYLKE er med i CLASS, men ikke i TABLE. Med MISSING-option

	Kjønn		
	I alt	1	2
	Antall	Antall	Antall
I alt	4	3	1
Sivil status			
1	1	1	-
2	1	1	-
3	2	1	1

Manglende verdier i analysevariable

Når det er *analysevariable* som mangler verdier, blir problemstillingen litt annerledes enn når grupperingsvariable er missing. Når grupperingsvariable mangler verdi, tas observasjonen *helt ut* av tabellen, den ignoreres. Hvis derimot en analysevariabel mangler verdi, tas ikke observasjonen *helt ut*, den vil telle med hvis vi teller opp antall observasjoner *uten* å knytte N til analysevariabelen. Men observasjonen vil *ikke* tas med i beregninger som utføres på analysevariabelen. Bemerk at beregninger ikke bare kan være sum, gjennomsnitt, standardavvik, varians etc, men også antall, N!

```
DATA test;
  INPUT sivstat $ 1 1
        inntekt 3 6 ;
  LABEL sivstat='Sivil status'
        inntekt='Inntekt' ;
  CARDS;
1 3000
1 1000
1 1500
2
2 2000
2 2000
3 1000
;

PROC TABULATE DATA=test FC='          ' NOSEPS F=8.;
  CLASS sivstat ;
  VAR  inntekt ;
  TABLE ALL='I alt' sivstat,
         N='Totalt antall'
         inntekt* (MEAN='Snitt' N='Antall med inntekt' SUM='Sum')
         /MISSTEXT=' ' RTS=14;
  TITLE 'OBS! Total N forskjellig fra antall med inntekt.';
RUN;
```

OBS! Total N forskjellig fra antall med inntekt.

	Inntekt			
	Totalt antall	Snitt	Antall med inntekt	Sum
I alt	7	1750	6	10500
Sivil status				
1	3	1833	3	5500
2	3	2000	2	4000
3	1	1000	1	1000

Her legger vi merke til at antall med inntekt *ikke* er det samme som totalt antall! Dette skyldes at én observasjon mangler inntekt. Denne blir dermed *ikke* med i N krysset med inntekt, men den blir med i den totale N, den som står alene. Legg merke til at snitt og sum er basert på de som har verdi for inntekt, missing holdes utenfor.



Som ovennevnte eksempler viser, det er svært viktig å ha nøye kjennskap til hvorvidt datamaterialet vi skal arbeide med inneholder manglende verdier. Bruk gjerne TABULATE og lag enkle frekvenstabeller (bruk evt. PROC FREQ) eller tell opp N og NMISS (bruk evt. PROC MEANS).

I noen tilfelle kan det være riktig å erstatte manglende verdier med 0, men det må vi bare gjøre hvis vi er 100 prosent sikre på at det er det korrekte. Et annet alternativ, kan være å imputere verdier. Dette vil si å erstatte missing med beregnede verdier basert på statistiske metoder.

Utfyllende om syntaksen i TABULATE

PROC TABULATE <options>
WHERE where-uttrykk;
CLASS grupperingsvariable </options> ;
CLASSLEV variable / **STYLE**=[stil-spesifikasjoner] | <PARENT> [stil-spesifikasjoner] ;
VAR analysevariable </options> ;
TABLE tabellspesifikasjon < / tabelloptions> ;
BY <sorteringsoptions> variable ;
FORMAT variabel format. <...variable format. > ;
FREQ variabel ;
KEYLABEL statistikknøkkelord ='tekst' <...statistikknøkkelord ='tekst'> ;
KEYWORD statistikknøkkelord | **ALL** / **STYLE**=[stil-spesifikasjoner] | <PARENT> [stil-spesifikasjoner] ;
LABEL variabel='tekst' <...variabel ='tekst'> ;
WEIGHT variabel ;
TITLE 'Overskrift' ;

Ikke alle setningene er nødvendige, minstekravet er **PROC TABULATE**, **CLASS** eller **VAR** og **TABLE**.

Options til PROC TABULATE-setningen:

ALPHA=verdi

Spesifiserer konfidensintervallet 100(1 - verdi). Gyldige verdier er 0.0001-0.9999.
 ALPHA=0.05 gir konfidensintervall på 95% (standard).

CONTENTS=

Endre innholdet i innholdsfortegnelse til angitt tekst. Kan brukes når vi lager html-filer sammen med CONTENTS=-option i ODS html.

CLASSDATA=

Navn på et annet SAS datasett som inneholder kombinasjoner av grupperingsvariable. Alle kombinasjoner kommer med i tabellen, med mindre vi velger EXCLUSIVE-option. Da selekteres bare de som er med i CLASSDATA-datasettet. Fin å bruke når vi ønsker å ha med alle kategorier i tabellen, selv om ikke alle har data i SAS datasettet. Alle grupperingsvariable som brukes i tabellen må være med i CLASSDATA-datasettet, fortrinnsvis med alle kombinasjoner av alle verdier. Kan også brukes til å lage seleksjonslister.

DATA= SAS datasett

Her oppgis navnet på SAS-datasettet tabellen skal lages fra. (Utelates DATA= vil TABULATE bruke det datasettet som er 'ferskest' i samme program eller interaktive SAS 'session'.)

EXCLNPWGHTS

Selekter kun observasjoner som har positive vekter. Se også WEIGHT-setningen.

EXCLUSIVE

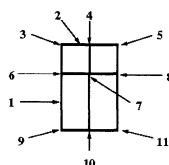
Brukes sammen med CLASSDATA=-option. Bare de verdier av grupperingsvariablene som er i SAS datasettet nevnt i CLASSDATA=-option selekteres.

FORMAT=formatnavn

Brukes hvis vi ønsker å angi felles formatering av tallene i tabellen, altså at alle kolonnene i utgangspunktet skal være like brede. FORMAT= kan forkortes til **F=** . FORMAT=7.2 gir en kolonnebredde på syv posisjoner, herav to desimaler. Bemerk at standard-formatet kan overstyres for enkeltkolonner ved at vi angir et annet format i selve TABLE-setningen. Utelates FORMAT= blir formatet BEST12.2 benyttet.

FC=' 11 tegn '

FC står for FormChar og brukes for å angi hvilke tegn tabellen skal rammes inn med. Det er 11 forskjellige møtepunkter i en tabellramme: Rekkefølgen er slik: 1. er vertikal strek, 2. er horisontal strek, 3. er øvre venstre hjørne, 4. er midten øverst, 5. er øvre høyre hjørne, 6. er midten til venstre, 7. er midt i tabellen, 8. er midten til høyre, 9. er nedre venstre hjørne, 10. er midten nederst og 11. er nedre høyre hjørne, se figur til venstre. FC=' -----' (1 blank og 10 streker) gir kun horisontale streker.



MISSING

Angis hvis vi vil at observasjoner som mangler verdi i *grupperingsvariable* likevel skal være med i tabellen. (Dersom MISSING ikke er med, blir observasjoner med missing i grupperingsvariablene utelatt.) Bemerk at MISSING ikke har noen innvirkning på behandlingen av missing i *analysevariable*.

NOSEPS

fjerner alle vannrette delelinjer i tabellinmaten. Linjene skrives da ut uten luft imellom, selv om vi har angitt blank som vannrett strek i FC=

NOTRAP

FPE (Floating point exception) recovery skrur av. Kan være gunstig for kjøretiden.

ORDER=rekkefølge

Angir hvilken rekkefølge verdiene til grupperingsvariable skal skrives ut i. Velger vi ORDER=FORMATTED sorteres de alfabetisk etter teksten i formatet som eventuelt brukes istedenfor etter koden. ORDER=FREQ sørger for at den verdi med flest observasjoner skrives først. ORDER=DATA gjør at verdiene beholder den rekkefølgen de har i datasettet. Den første forekomsten av verdiene bestemmer rekkefølgen. Dette gjelder ikke når vi bruker CLASSDATA- eller PRELOADFMT-option. Da blir rekkefølgen som definert i CLASSDATA-datasettet eller formatet til variabelen som har PRELOADFMT-option. Standardverdi er ORDER=UNFORMATTED (= INTERNAL) som skriver ut verdiene i sortert rekkefølge.

OUT=

Legger tabellen ut på SAS datasett. Følgende variable lages:

- Variable definert med BY-setningen
- Variable definert i CLASS-setningen
- _TYPE_, viser sammenhengen mellom grupperingsvariablene
- _PAGE_, den logiske siden som inneholder observasjonen
- _TABLE_, tabellnummeret observasjonen hører til
- Statistikkvariable, utførte beregninger

SAS datasettet som lages vil kunne være vanskelig å bruke videre. Det avhenger av hvor kompliserte TABLE-setninger vi har brukt. Vi får et SAS datasett som minner om de vi kan lage ved å aggregere i PROC MEANS. Hver BY-variabel og grupperingsvariabel kommer ut på det nye SAS datasettet, og det lages en variabel pr. analysevariabel og statistikk.

STYLE=

Brukes i forbindelse med html-generering for å angi standard utseende for tabellen.

THREADS

Åpner for parallellprosessering av dataene over flere disketter samtidig. Dette vil senke prosesseringstiden.

VARDEF=divisor

Angir divisor som brukes til å regne ut varianser. Gyldige verdier er:

DF	N-1 skal brukes som divisor (standard)
N	N skal brukes som divisor
WDF	Summen av vektene - 1 skal brukes som divisor
WEIGHT	Summen av vektene skal brukes som divisor

Andre options som ikke er beskrevet her: QMARKERS, QMETHOD, QNTLDEF og TRAP.

Setninger til PROC TABULATE:

WHERE-setningen

Selekterer hvilke observasjoner det skal lages tabeller for. Bare de som tilfredstiller WHERE-betingelsen tas med.

BY-setningen

BY-setningen brukes på samme måte i TABULATE som i de andre SAS-prosedyrene. Bruker vi BY-setningen, vil vi få én tabell for hver verdi av BY-variabelen. Hvis vi bruker BY-setningen må SAS-datasettet være sortert på forhånd (med mindre vi har med sorteringsoption NOTSORTED).

Istedenfor å bruke BY i TABULATE, vil det som oftest lønne seg å bruke sidedimensjonen i TABLE-setningen. Denne gjør samme nytte som BY, men krever **ikke** at SAS-datasettet er sortert på forhånd.



CLASS-setningen

I CLASS-setningen lister vi opp grupperingsvariablene, klassifiseringsvariablene. Grupperingsvariable er variable med diskrete verdier. Disse er som oftest definert som karaktervariable (av type *character*), men de kan også være numeriske (av type *numeric*). Numeriske, (dvs. kontinuerlige) variable bør grupperes ved hjelp av et format for å kunne brukes som grupperingsvariable.

Vær oppmerksom på at observasjoner med missing i en eller flere av grupperingsvariablene utelates fra tabellen såfremt vi ikke har med MISSING i selve PROC TABULATE-setningen. Dette gjelder for alle variablene i CLASS-setningen, selv om de ikke brukes i tabellen.

Vi kan ha flere CLASS-setninger i samme TABULATE-program.

Options til CLASS-setningen:

ASCENDING

Grupperingsvariablene skrives ut i stigende rekkefølge, dette er standard.

DESCENDING

Grupperingsvariablene skrives ut i synkende rekkefølge.

EXCLUSIVE

Brukes sammen med PRELOADFMT=option. Bare de verdier av grupperingsvariablene som er i de nevnte SAS-formatene selekteres.

GROUPINTERNAL

Formater til grupperingsvariablene hektes på etter at SAS har gruppert dem. Dette kan spare noe maskintid. Kan brukes når formatene ikke grupperer forskjellige variabelverdier. Når vi bruker formater til å slå sammen grupper, f.eks. fra alder til 5-årige aldersgrupper, bør vi ikke bruke GROUPINTERNAL. Vi må også være oppmerksomme på restgrupperinger definert med OTHER i PROC FORMAT. Har disse forskjellige verdier, vil hver verdi komme på hver sin linje (ev. kolonne) i tabellen.

MISSING

Sørger for at MISSING behandles som gyldig verdi for grupperingsvariablene. Påvirker ingen statistiske beregninger.

MLF

Angir at vi kan bruke SAS formater med overlappende verdier. Krever at formatene er definert som MULTILABEL i PROC FORMAT. Gjør at samme variabel kan grupperes på forskjellige måter i samme format, og vi kan få dem ut i en tabell.

ORDER=rekkefølge

Angir hvilken rekkefølge verdiene til grupperingsvariable skal skrives ut i. Velger vi ORDER=FORMATTED blir de sortert alfabetisk etter teksten i formatet som eventuelt brukes istedenfor etter koden. ORDER=FREQ sørger for at den verdi med flest observasjoner skrives først. ORDER=DATA gjør at verdiene beholder den rekkefølgen de har i datasettet. Den første forekomsten av verdiene bestemmer rekkefølgen. Dette gjelder ikke når vi bruker CLASSDATA- eller PRELOADFMT-option. Da blir rekkefølgen som definert i CLASSDATA-datasettet eller formatet til variabelen som har PRELOADFMT-option. Standardverdi er ORDER=UNFORMATTED som skriver ut verdiene i sortert rekkefølge.

PRELOADFMT

Vi bruker PRELOADFMT for å si at vi ønsker å beholde rekkefølgen formatet fikk da det ble lagret. Med denne option og NOTSORTED-option til VALUE-setningen i PROC FORMAT, kan vi sikre at vi får linjene og kolonnene i den riktige rekkefølgen i forspalte og hodet i tabellen. For at denne skal virke må option ORDER=data være med i PROC TABULATE-setningen, se eksempel side 8.

STYLE=stilspesifikasjoner

Brukes i forbindelse med html-generering for å angi standard utseende for tabellen.

CLASSLEV-setningen

CLASSLEV-setningen kan vi bruke i forbindelse med utskrift til html-filer til å endre på utseende til grupperingsvariable i forspalten eller kolonnehoder. Vi kan ha flere CLASSLEV-setninger i samme program. Kan f.eks. brukes til å gi forskjellig utseende til forskjellige nivåer i forspalten.

FORMAT-setningen

FORMAT-setningen brukes på samme måte i TABULATE som i de andre SAS-prosedyrene. Ved hjelp av FORMAT-setningen kan vi erstatte variabelverdier med mer forståelig *tekst* (f.eks at verdien 1 erstattes med teksten Menn og verdien 2 erstattes med teksten Kvinner).

Vi kan også bruke FORMAT-setningen til å *gruppere* variabelverdier (lage aldersgrupper, innteksgrupper etc). Det sistnevnte kan være spesielt nyttig i TABULATE, for på den måten kan vi bruke variable som tradisjonelt er analysevariable (alder, inntekt og andre numeriske, gjerne kontinuerlige variable) som grupperingsvariable.

FREQ-setningen

Hvis hver observasjon på datasettet skal telle for mer enn 1, kan vi bruke FREQ-setningen. I FREQ-setningen kan vi da angi variabelen hvis verdi inneholder det antall observasjoner denne observasjonen skal telle som. Dette betyr at N også økes med verdien til FREQ-variabelen.

FREQ trunkerer alle desimaler i FREQ-variabelen før den brukes. FREQ regner altså kun med heltall. Skal vi veie observasjonene i henhold til en variabel bør vi bruke WEIGHT istedenfor FREQ.

KEYLABEL-setningen

Vil vi ha egne tekster på statistikknøkkelordene som brukes i tabellen kan vi bruke KEYLABEL. Nøkkelord er SUM, MEAN, N etc. KEYLABEL MEAN='Gjennomsnitt'; Bemerk at dette også kan gjøres i TABLE-setningen. Vi kan også tekste totalvariabelen ALL med KEYLABEL.

KEYWORD-setningen

KEYWORD-setningen kan vi bruke i forbindelse med utskrift til html-filer til å endre på utseende til statistiske nøkkelord og ALL i forspalten eller kolonnehoder. Vi kan ha flere KEYWORD-setninger i samme program.

LABEL-setningen

Hvis variablene vi bruker har forklarende tekster ('labler'), er det disse som blir skrevet ut i tabellen. Dersom variablene *ikke* har labler, eller vi ønsker å *endre* lablene for anledningen, kan vi gjøre dette med LABEL-setningen. LABEL kjonn = 'Kjønn'; Bemerk at dette også kan gjøres i TABLE-setningen.

TABLE-setningen

I TABLE-setningen beskrives tabellen i detalj. Derfor kan TABLE-setningen ofte bli ganske lang. Del den gjerne over flere linjer, det øker forståeligheten og sjansen for å oppdage feil i tide. Generelt kan TABLE-setningen skrives slik:

```
TABLE sideuttrykk,
      forspalteuttrykk,
      hodeuttrykk
      /options ;
```

En tabell i TABULATE ha opptil 3 dimensjoner: side, forspalte og hode. Uttrykkene som beskriver side-, forspalte- og hode-dimensjonen skilles fra hverandre med komma. Det første uttrykket definerer siden, det neste forspalten og det siste hodet. Dersom man har med 2 uttrykk, blir det første forspalte og det siste hode. Med bare 1 uttrykk får tabellen bare hode, ingen forspalte. Side, forspalte og hode defineres etter samme prinsipper.

Uttrykkene som definerer side, forspalte og hode kan inneholde følgende elementer:

Analysevariable	disse <i>må</i> være definert i VAR-setningen
Grupperingsvariable	disse <i>må</i> være definert i CLASS-setningen
ALL	egen grupperingsvariabel for totaler
Statistikk-nøkkelord	CSS, CV, KURTOSIS, MAX, MEAN, MIN, N, NMISS, PCTN, PCTSUM, PROBT, RANGE, SKEWNESS, STD, STDERR, SUM, SUMWGT, USS, T og VAR, for nærmere beskrivelse, se side 74.
	Videre kan disse beregnes:
LCLM	Nedre grense for konfidensintervall (definert med ALPHA-opsjon i TABULATE-setningen)
UCLM	Øvre grense for konfidensintervall (definert med ALPHA-opsjon i TABULATE-setningen)
Q3	Øvre kvartil (75. prosentil = P75)

MEDIAN	Medianverdi (50. prosentil = P50)
Q1	Nedre kvartil (25. prosentil = P25)
QRANGE	Avstanden mellom øvre og nedre kvartil, (Q3-Q1)
P1	1. prosentil
P5	5. prosentil
P10	10. prosentil
P90	90. prosentil
P95	95. prosentil
P99	99. prosentil

For prosentuering kan vi spesifisere hvilken måte det skal prosentures på med disse variantene: COLPCTN, COLPCTSUM, PAGEPCTN, PAGEPCTSUM, REPPCTN, REPPCTSUM, ROWPCTN og ROWPCTSUM. Hvis ingen av disse gir ønsket prosentuering, må vi bruke PCTN eller PCTSUM med <> for å spesifisere prosentueringen.

Formater	bestemmer tallenes utseende og kolonnebredde (antall desimaler etc.)
Tekst	erstatte variabelnavn eller statistikk-nøkkelord med tekst
Dessuten kan følgende operatører brukes i TABLE-setningen:	
,	komma skiller side-, forspalte- og hode-dimensjonene i tabellen
*	stjerne krysser elementer (setter <i>innenfor</i> hverandre)
	blank i forspalten: setter elementer <i>etter hverandre</i> i hodet: setter elementer <i>ved siden av hverandre</i>
Dette kalles også <i>konkatenering</i>	
()	parenteser grupperer deler av uttrykk
<>	klammer spesifiserer nevner i prosentuering
=	likhetstegn 1. for å angi tekst til en variabel eller et statistikk-nøkkelord 2. for å angi et format (f.eks ...*F=8.2)

Elementene i tabellen kan krysses, settes etter hverandre og grupperes i de kombinasjoner vi måtte ønske. Når vi krysser to grupperingsvariable, vil de fordeles hierarkisk med den først nevnte som høyeste (forspalten: ytterste, hode: øverste) nivå. Krysses en grupperingsvariabel med en analysevariabel, vil det utføres statistiske beregninger på analysevariabelen fordelt etter grupperingsvariabelen.

Kryssing av to grupperingsvariable i forspalten: fylke * kjonn

Østfold	Menn	16038
	Kvinner	18995
Akershus	Menn	23070
	Kvinner	29629

Konkatenering av grupperingsvariable: fylke kjonn

Østfold	35033
Akershus	52699
Menn	39108
Kvinner	48624

Kryssing og gruppering av grupperingsvariable: fylke * (ALL kjonn)

Østfold	ALL	35033
	Menn	16038
	Kvinner	18995
Akershus	ALL	52699
	Menn	23070
	Kvinner	29629

En TABLE-setning definerer én tabell, men vi kan ha flere TABLE-setninger i én TABULATE. Vi må da være oppmerksom på at alle tabellene får samme overskrift og at missing i grupperingsvariable ikke tas med, selv om variabelen ikke brukes i TABLE-setningen (med mindre MISSING-option er med i PROC TABULATE-setningen).

Options til TABLE-setningen:

BOX= verdi | {<label= 'tekst'> <STYLE=<stil-element-navn>

[stil-attributt-navn=stil-attributt-verdi<... stil-attributt-navn=stil-attributt-verdi>]>}

Angir tekst for forspaltehode (tekst over linjene, 'boksen' i øverste venstre hjørne). Som verdi kan vi velge en tekst angitt i fnutter (f.eks BOX='Omnibus 2004') eller et variabelnavn (gir variabelens label som forspaltehode). Vi kan også skrive BOX=_PAGE_ men dette kan vi bare gjøre hvis vi har med sidedimensjonen i tabellen. Det som skrives i 'boksen' er da verdien av grupperingsvariabelen som utgjør sidedimensjonen. Vi kan også angi stilspecificasjoner for html-utskrift her.

CONDENSE

Hvis vi har en tabell med 3 dimensjoner (mao en tabell som også har en *sidedimensjon*), kan vi bruke CONDENSE til å få skrevet ut så mange av 'siden' som mulig på ett ark. Dette for å begrense papirmengden.

Hvis vi definerer en tabell som er bredere enn linjebredden tillater, vil SAS dele tabellen og hver bit vil bli skrevet ut på hver sin side. Med CONDENSE vil bitene skrives ut (under hverandre) på samme side dersom det er plass til det. CONDENSE har ingen effekt på sideskift generert ved bruk av BY-setningen

CONTENTS=tabell-referanse

Kan brukes for å angi en referanse som skal bruke til å bygge opp en liste over tabeller når vi lager html-filer.

FORMAT_PRECEDENCE=PAGE|ROW|COL

Angir hvilken dimensjon som skal ha forrang når formater for celler som er definert i flere dimensjoner i TABLE-setningen. Standard er COL.

FUZZ=verdi

Kan brukes til å eliminere ubetydelige verdier. Verdier mindre enn FUZZ-verdien for analysevariable blir tolket som 0 i beregninger og utskrift.

INDENT=verdi

Angir at vi ønsker at underliggende nivåer i rekke-dimensjonen skal skrives ut under hverandre med den oppgitte verdi som innrykk. Dette istedenfor at de settes ved siden av hverandre.

MISSTEXT='tekst'

Istedenfor at manglende verdier (missing) skrives ut i tabellinnmaten som punktum, kan vi selv velge en tekst for å angi missing. Med MISSTEXT='- ' vil missing i tabellen bli skrevet ut som bindestrek.

NOCONTINUED

Brukes for å slippe å få fortsettelsesteksten (CONTINUED) på utskriften fra andre side og utover.

PRINTMISS

PRINTMISS kan brukes når vi *krysser* grupperingsvariable. Med PRINTMISS blir *alle* verdiene til den innerste grupperingsvariabelen skrevet ut *for hver verdi* av den ytre grupperingsvariabelen, selv om tabellcellen for den aktuelle kombinasjonen er missing.

I eksempelet under finnes det ingen observasjoner med fylke=02 og kjønn=1, heller ingen med fylke=03 og kjønn=2. Uten PRINTMISS vil det ikke lages kolonner for disse kombinasjonene. Med PRINTMISS lages det kolonner også for de kombinasjoner av fylke og kjønn som ikke er på datasettet.

```

DATA test;
  INPUT fylke    $ 1 2
        kjonn    $ 4 4 ;
  CARDS;
01 1
01 2
02 2
03 1
;

PROC TABULATE DATA=test FC='          ' NOSEPS F=8.;
  CLASS fylke kjonn ;
  FORMAT fylke $fylke.
        kjonn $kjonn. ;
  TABLE fylke=' '*kjonn=' '*N='Antall'
        ;
  TITLE 'Uten PRINTMISS.';

PROC TABULATE DATA=test FC='          ' NOSEPS F=8.;
  CLASS fylke kjonn ;
  FORMAT fylke $fylke.
        kjonn $kjonn. ;
  TABLE fylke=' '*kjonn=' '*N='Antall'
        /PRINTMISS;
  TITLE 'Med PRINTMISS.';
RUN;

```

Uten PRINTMISS.

Østfold		Akershus		Oslo	
Menn	Kvinner	Kvinner	Menn	Menn	Kvinner
Antall	Antall	Antall	Antall	Antall	Antall
1	1	1	1	1	1

Med PRINTMISS.

Østfold		Akershus		Oslo	
Menn	Kvinner	Menn	Kvinner	Menn	Kvinner
Antall	Antall	Antall	Antall	Antall	Antall
1	1	.	1	1	.

ROW=

Standard verdi er ROW=CONSTANT, alternativet er ROW=FLOAT. For å forstå hvordan vi kan bruke ROW= må vi først være klar over det faktum at tilgjengelig ferspaltebredde *fordeles likt* mellom de elementene som utgjør ferspalten. Hvis ferspalten består av flere elementer kryssset, vil hvert element få like stor plass i ferspalten. Hvis ferspalten f.eks er fylke*kjonn, og tilgjengelig ferspaltebredde er 20 tegn, vil fylke og kjønn få 10 tegn hver til rådighet. (Dette er en stor svakhet: Hvis vi skal lage en bred tabell, må vi ofte knipe inn ferspaltebredden med RTS= . Hvis ferspalten vår da er fylke kryssset med kjønn og RTS=28, hadde det vært ønskelig å kunne fordele ferspaltebredden med 18 tegn til fylke og 10 tegn til kjønn, 'Sogn og Fjordane' er mye bredere enn 'Kvinner'. Slik det er, får fylke og kjønn 14 tegn hver, hvilket fører til at 'Sogn og Fjordane' skrives ut på to linjer.)

ROW= kan ikke løse problemet i eksempelet med fylke og kjønn. Men ROW= kan benyttes dersom det innerste elementet av ferspalten er blank! Dette kan f.eks forekomme når vi krysser en variabel med en statistisk beregning og vi knytter blank tekst til statistikk-nøkkelordet (f.eks FYLKE*N=" ,). I dette tilfelle kan vi unngå at ferspalten fordeles likt mellom fylke og N (antall). Med ROW=FLOAT vil nemlig ferspaltebredden fordeles likt mellom de *ikke-blanke* elementer, så hvis ferspaltebredden er 20 tegn vil fylke få alle 20, i motsetning til med ROW=CONSTANT der fylke vil få 10 og antall (selv om den er blank) vil få 10.

```

DATA test;
  INPUT fylke    $ 1 2
        kjonn    $ 4 4 ;
  CARDS;
14 1
14 2
15 2
16 1
;

PROC TABULATE DATA=test FC='          ' NOSEPS F=8.;
  CLASS fylke kjonn ;
  FORMAT fylke $fylke.
        kjonn $kjonn. ;
  TABLE fylke=' '*N=',
        kjonn=' '
        /RTS=18 ;
  TITLE 'ROW=CONSTANT (standard) ' ;

PROC TABULATE DATA=test FC='          ' NOSEPS F=8.;
  CLASS fylke kjonn ;
  FORMAT fylke $fylke.
        kjonn $kjonn. ;
  TABLE fylke=' '*N=',
        kjonn=' '
        /RTS=18 ROW=FLOAT;
  TITLE 'ROW=FLOAT ' ;
RUN;

```

ROW=CONSTANT (standard)		
	Menn	Kvinner
Sogn og Fjordane	1	1
Møre og Romsdal	.	1
Sør-Trøndelag	1	.

ROW=FLOAT		
	Menn	Kvinner
Sogn og Fjordane	1	1
Møre og Romsdal	.	1
Sør-Trøndelag	1	.

RTS=antall tegn

RTS (står for Row Title Space) er det antall tegn som settes av til forspalten. Hvis vi ikke angir noen RTS, blir forspaltebredden automatisk en fjerdedel av linjebredden.

STYLE=

Spesifiserer stiler for bruk på html-tabeller.

STYLE_PRECEDENCE=PAGE|ROW|COL

Angir hvilken dimensjon som skal ha forrang når ods-stiler for celler som er definert i flere dimensjoner i TABLE-setningen. Standard er COL.

TITLE-setningene

Vi har mulighet for inntil 10 linjer overskrift, TITLE1, TITLE2, ..., TITLE10 (TITLE betyr altså TITLE1). Overskriften må stå i fnutter. Her har vi lov til å bruke alle tegn, øøå inkludert.

VAR-setningen

I VAR-setningen lister vi opp analysevariablene våre. Analysevariablene er de det beregnes statistikk på (SUM, MEAN, MAX, MIN etc). Analysevariablene *må* være numeriske. Manglende verdier for analysevariable teller ikke med i beregningene (bortsett fra NMISS som er antall observasjoner med missing).

Options til VAR-setningen:

STYLE=stilspesifikasjoner

Brukes i forbindelse med html-generering for å angi standard utseende for tabellen.

WEIGHT=vektvariabel

Angir vektvariabel for analysevariablene. WEIGHT-option i VAR-setningen overstyres WEIGHT-setningen.

WEIGHT-setningen



For å veie dataene i tabellen bruker vi WEIGHT-setningen. Vi må da ha en variabel på datasettet som inneholder vekten (oppblåsingsfaktoren). Vektvariabelen kan gjerne være et desimaltall.

Vær oppmerksom på at N aldri veies, heller i ikke ved prosentregning (PCTN). Det samme gjelder for NMISS. Hvis vi skal ha veide prosenter, må vi bruke PCTSUM.



Bemerk at WEIGHT-setningen brukes annerledes i TABULATE enn i PROC FREQ! Bruker vi WEIGHT-setningen i PROC FREQ blir N og PCTN veid, i motsetning til i TABULATE der disse *ikke* blir veid.

Når vektvariabelen er negativ, får den verdien 0 og observasjonen telles med i antallet. Når verdien er 0 telles observasjonen med i antallet (N). Dette kan overstyres med PROC TABULATE-option EXCLNPWGHTS som ekskluderer disse observasjonene fra tabellen. Er vektvariabelen MISSING, ekskluderer observasjonen fra tabellen. WEIGHT-setningen overstyres av WEIGHT-option i VAR-setningen.

Feilmeldinger i TABULATE

I vår omgang med TABULATE kan det hende vi får en feilmelding i SASLOG'en istedenfor en tabell. De fleste feilmeldingene er forståelige, f.eks at man bruker en variabel i TABLE-setningen uten å ha definert den i CLASS- eller VAR-setningen. Imidlertid er det noen feilmeldinger som kan være vanskelige å tolke:

Finner ikke variabelen som brukes i tabellen

Hvis vi glemmer å nevne en variabel vi bruker i TABLE-setningen (eller feilstaver den i TABLE-setningen) i CLASS-, VAR-, eller WEIGHT-setningen får vi denne type feilmelding.

```
ERROR: The type of name (kjonnn) is unknown.
```

Statistiske beregninger uten analysevariabel

Skal vi gjøre statistiske beregninger, må vi gjøre dette på analysevariable. I programmet under har vi definert analysevariabelen inntekt i VAR-setningen, men vi har glemt å ta den med i TABLE-setningen:

```
PROC TABULATE DATA=bef FC='-----' NOSEPS F=6. ;
  CLASS landsdel kjonn;
  VAR innbyggere ;
  TABLE (ALL landsdel), (ALL kjonn)*SUM;
  TITLE 'Statistiske beregninger uten analysevariabel.';
RUN;
```

```
ERROR: Statistic other than N was requested without analysis variable in the
following nesting : All * All * Sum.
ERROR: Statistic other than N was requested without analysis variable in the
following nesting : All * Kjonnn * Sum.
ERROR: Statistic other than N was requested without analysis variable in the
following nesting : Landsdel * All * Sum.
ERROR: Statistic other than N was requested without analysis variable in the
following nesting : Landsdel * Kjonnn * Sum.
```

Ved å ta med analysevariabelen inntekt i TABLE-setningen, vil programmet gå bra:

```
TABLE (ALL landsdel), (ALL kjonn)*innbyggere*SUM;
```

Statistiske beregninger i både forspalte og hode

Vi kan ikke ha statistiske beregninger i både forspalte og hode. En TABLE-setning som dette, vil derfor gå galt og gi oss en feilmelding: TABLE (ALL landsdel)*N, (ALL kjonn)*N;

```
ERROR: There are multiple statistics associated with a single table cell in the
following nesting : All * N * All * N.
ERROR: There are multiple statistics associated with a single table cell in the
following nesting : All * N * Kjonnn * N.
ERROR: There are multiple statistics associated with a single table cell in the
following nesting : Landsdel * N * All * N.
ERROR: There are multiple statistics associated with a single table cell in the
following nesting : Landsdel * N * Kjonnn * N.
```

Her må vi ta bort N enten fra forspalten eller fra hodet for å få laget tabellen.

Ufullstendig nevner i prosentuering

Ved forsøk på å prosentuerer uten å ha definert *hele* nevneren, vil vi få feil. Med denne TABLE-setningen går det galt: TABLE (ALL landsdel), (ALL kjønn)*PCTN<landsdel>; SAS finner ut at vi ikke har definert hele nevneren og serverer oss denne meldingen:

```
ERROR: PctN base is not in table. A PctN crossing has no denominator.
```

Her må nevneren omfatte hele forspalten (med det mest aggregerte uttrykket sist): PCTN<landsdel all> for å få laget tabellen, eller vi kan eventuelt bruke den nye måten å lage kolonneprosenter på: COLPCTN.

Ingen data å lage tabell av

Hvis en observasjon mangler verdi for en av grupperingsvariablene, utelates som kjent hele observasjonen fra tabellen (såfremt vi ikke har med MISSING-option i TABULATE-setningen). Mangler alle observasjonene på datasettet verdi for minst en av grupperingsvariablene, vil altså alle observasjonene holdes utenfor tabellen og det blir ingen observasjoner igjen å lage tabell for. Denne meldingen vil da komme i SASlog:

```
WARNING: A class, frequency, or weight variable is missing on every observation.
```

Ved å ta med MISSING-option i TABULATE-setningen vil det bli laget en tabell. Denne bør sjekkes nøye for å finne ut hvilken effekt de manglende verdier har på resultatet i tabellen.

Kort om syntaksen i REPORT

Dette er en vanlig oppbygging av et REPORT-program:

```
PROC REPORT DATA= SAS datasett HEADLINE HEADSKIP NOWINDOWS MISSING;
  COLUMN variable til rapporten;
  DEFINE grupperingsvariabel forspalte / GROUP ;
  DEFINE grupperingsvariabel hode/ ACROSS ;
  DEFINE analysevariabel / ANALYSIS ;
  RBREAK BEFORE / SUMMARIZE SKIP ;
  TITLE 'Overskrift ' ;
```

Programmet forklares slik:

PROC REPORT-setningen:

I denne setningen oppgir vi hvilket SAS datasett som skal være utgangspunkt for rapporten (**DATA=**). Videre sier vi at vi skal ha en strek (**HEADLINE**) og en blank linje (**HEADSKIP**) under kolonneoverskriftene. **NOWINDOWS** angir at rapporten skal legges i Output-vinduet og ikke i et eget Report-vindu. **MISSING** betyr at observasjoner med manglende verdier (missing) i grupperingsvariablene tas med i tabellen.

COLUMN-setningen:

I **COLUMN**-setningen lister vi opp variablene som skal brukes i rapporten i den rekkefølgen vi ønsker at de skal komme ut i rapporten.

DEFINE -setningene:

Her lager vi en **DEFINE**-setning for hver variabel som skal være med i rapporten. Vi angir også bruksmåte for variablene her. **GROUP** angir gruppering i forspalten, **ACROSS** gruppering i hodet og **ANALYSIS** at variabelen skal brukes til statistikk, for eksempel sum (**SUM**) eller gjennomsnitt (**MEAN**). I tillegg kan vi lage nye, beregnede variable. Disse må defineres i **COLUMN**-setningen og ha en egen **DEFINE**-setning der bruksmåten angis med **COMPUTED**.

RBREAK-setningen:

Med **RBREAK** sørger vi for at det lages en sumlinje. Denne kan plasseres øverst (**BEFORE**) eller nederst i rapporten (**AFTER**).

Rapportene defineres ut fra kolonner. Hver variabel får sin kolonne, unntatt **ACROSS**-variable. De får en kolonne for hver verdi variabelen inneholder. Alle kolonner må defineres i **COLUMN**-setningen.

Mens tabellene i **TABULATE** bygges opp ved å definere side, forspalte og kolonner som dimensjoner som skilles med komma, forholder **REPORT** seg kun til kolonner. Den første kolonnen (evt. de to første) vil vanligvis bli en forspalte, mens resten vil bli kolonner. En typisk tabell i **REPORT** krever flere programlinjer enn en i **TABULATE**.

Hvordan lage enkle tabeller med REPORT

Vi kan altså også bruke REPORT til å lage tabeller. REPORT krever ofte noe mer skriving før vi får ut tabeller som vi ønsker oss. Vi starter med en enkel aku-utlisting der vi velger ut tre næringer. For disse ønsker vi å liste ut enkelobservasjoner, omtrent som vi ville gjort med PROC PRINT. Programmet kan se slik ut:

```
PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE naering IN('23','24','25');
  COLUMN naering landsdel kjonn vekt1996-vekt1999;
  TITLE 'Rapport 1. Utlisting av næring 23-25' ;
RUN;
```

Vi har ikke brukt en eneste DEFINE-setning og SAS vil da velge bruksmåte DISPLAY som brukes til å liste ut enkeltobservasjoner slik som her, vi legger merke till at variablene skrives ut i den rekkefølgen de er nevnt i COLUMN-setningen:

```
Rapport 1. Utlisting av næring 23-25

      L
      a
      n
      d
      s
      j
Næ d ø
ri e n
ng l n  Vekt 1996  Vekt 1997  Vekt 1998  Vekt 1999
-----
23  1  1      33331      34346      34241      38811
24  1  2      55525      56723      56352      53100
25  1  2      45436      46664      47363      49323
25  3  1      55436      56664      57363      57723
25  4  1      55436      56664      57363      56323
```

Hver observasjon i SAS datasettet får en rad i utlistingen. Når vi ikke brukte noen DEFINE-setninger kunne vi heller ikke tilpasse tabellen noe. Ved å ta i bruk DEFINE-setninger kan vi pynte på tabellens utseende. Her har vi valgt å angi hvor brede kolonnene i tabellen skal være (WIDTH) og for den første har sagt at den skal skrives ut uten noen blanke først (SPACING=0). Standard er at det skrives ut to blanke før en kolonne skrives ut (SPACING=2).

```
PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE naering IN('23','24','25');
  COLUMN naering landsdel kjonn vekt1996-vekt1999;
  DEFINE naering / DISPLAY WIDTH=6 SPACING=0;
  DEFINE landsdel / DISPLAY WIDTH=8 ;
  DEFINE kjonn / DISPLAY WIDTH=5 ;
  DEFINE vekt1996 / DISPLAY WIDTH=10 ;
  DEFINE vekt1997 / DISPLAY WIDTH=10 ;
  DEFINE vekt1998 / DISPLAY WIDTH=10 ;
  DEFINE vekt1999 / DISPLAY WIDTH=10 ;
  TITLE 'Rapport 2. Utlisting av næring 23-25' ;
RUN;
```

Utlistingen er nå blitt noe penere:

Rapport 2. Utlisting av næring 23-25						
Næring	Landsdel	Kjønn	Vekt 1996	Vekt 1997	Vekt 1998	Vekt 1999
23	1	1	33331	34346	34241	38811
24	1	2	55525	56723	56352	53100
25	1	2	45436	46664	47363	49323
25	3	1	55436	56664	57363	57723
25	4	1	55436	56664	57363	56323

Skal vi liste ut observasjoner bruker vi vanligvis heller PROC PRINT. Vi skal nå lage en tabell med gruppering både i forspalten (GROUP) og i hodet (ACROSS). Vi må definere at vi ønsker næring i forspalten og kjønn i hodet. Videre sier vi at vi ønsker å summere analysevariablene (vekt1996-vekt1999).

```

PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  COLUMN næring kjønn vekt1996-vekt1999;
  DEFINE næring / GROUP WIDTH=6 SPACING=0;
  DEFINE kjønn / ACROSS WIDTH=5 ;
  DEFINE vekt1996 / ANALYSIS SUM WIDTH=10 ;
  DEFINE vekt1997 / ANALYSIS SUM WIDTH=10 ;
  DEFINE vekt1998 / ANALYSIS SUM WIDTH=10 ;
  DEFINE vekt1999 / ANALYSIS SUM WIDTH=10 ;
  TITLE 'Rapport 3. Gruppering etter næring og kjønn' ;
RUN;

```

Utskriften blir slik, alle med samme verdi for næring er gruppert sammen. Tilsvarende gjelder for kjønn. Vi har nå tatt med alle næringer i SAS datasettet. Vi ser at det lages en kolonne for hvert kjønn og en kolonne for hver vekt-variabel.

Rapport 3. Gruppering etter næring og kjønn						
Næring	Kjønn		Vekt 1996	Vekt 1997	Vekt 1998	Vekt 1999
	1	2				
01	.	1	45436	46664	47363	48323
02	.	.	33331	34346	34241	38811
05	.	1	99765	102147	98592	103735
10	.	2	90872	93328	94726	95646
11	.	1	33331	34346	34241	38811
12	.	1	45436	46664	47363	41323
13	.	1	33331	34346	34241	38811
14	.	.	55525	56723	56352	53100
15	.	.	44240	45424	42240	45635
16	.	1	120872	124328	122726	122746
17	.	1	33331	34346	34241	48811
18	.	.	45436	46664	47363	48323
19	.	1	33331	34346	34241	38811
20	.	.	55525	56723	56352	53100
21	.	.	44240	45424	42240	45635
22	.	1	45436	46664	47363	49323
23	.	1	33331	34346	34241	38811
24	.	.	55525	56723	56352	53100
25	.	2	156308	159992	162089	163369
26	.	1	33331	34346	34241	48811
27	.	.	113302	114358	112025	110100
28	.	1	45436	46664	47363	.
29	.	1	33331	34346	34241	38811
30	1	.	99758	102358	99673	106082
31	.	.	44240	45424	42240	45635
32	.	2	99956	99812	102902	115944
33	.	.	111050	113446	112704	115200
34	.	1	44240	45424	42240	49635
35	.	1	45436	46664	47363	48323
36	.	1	33331	34346	34241	49811
37	.	1	45436	46664	47363	48323
40	.	1	33331	34346	34241	38811
41	.	.	55525	56723	56352	56100
45	1	1	78473	79059	77561	84617

Denne tabellen ble ganske stor og vi kan gjøre den mindre ved å slå sammen næringer. Vi bruker PROC FORMAT og lager et næringsformat. Dette bruker vi så i en FORMAT-setning i REPORT. Vi bruker også et tidligere definert kjønnsformat.

```

PROC FORMAT ;
  VALUE $nar
  '01'          = 'Jordbruk, jakt og viltstell'
  '02'          = 'Skogbruk'
  '05'          = 'Fiske, fangst og fiskeoppdrett'
  '11'          = 'Utv.av råolje og naturgass,inkl.tj.'
  '10','12'-'14' = 'Bergverksdrift'
  '15'-'37'    = 'Industri'
  '40'-'41'    = 'Kraft- og vannforsyning'
  '45'          = 'Bygge- og anleggsvirksomhet'
;
RUN;
PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  COLUMN naering kjonn vekt1996-vekt1999;
  DEFINE naering / GROUP WIDTH=6 SPACING=0;
  DEFINE kjonn / ACROSS WIDTH=5 ' ';
  DEFINE vekt1996 / ANALYSIS SUM WIDTH=10 ;
  DEFINE vekt1997 / ANALYSIS SUM WIDTH=10 ;
  DEFINE vekt1998 / ANALYSIS SUM WIDTH=10 ;
  DEFINE vekt1999 / ANALYSIS SUM WIDTH=10 ;
  FORMAT kjonn $kjonn.
         naering $nar.;
  TITLE 'Rapport 4. Tekstet gruppering etter næring og kjønn' ;
RUN;

```

Tabellen ble ikke spesielt pen, til det er noen kolonner for smale. Teksten ble kuttet for næringsgruppene og den ble delt for kjønn:

Rapport 4. Tekstet gruppering etter næring og kjønn							
Næring	Kvinner	Menn	Uoppgitt	Vekt 1996	Vekt 1997	Vekt 1998	Vekt 1999
Bergve	1	4	.	225164	231061	232682	228880
Bygge-	.	1	1	78473	79059	77561	84617
Fiske,	1	1	.	99765	102147	98592	103735
Indust	13	16	1	1416422	1448832	1436044	1484339
Jordbr	.	1	.	45436	46664	47363	48323
Kraft-	1	1	.	88856	91069	90593	94911
Skogbr	1	.	.	33331	34346	34241	38811
Utv.av	.	1	.	33331	34346	34241	38811

Vi vil nå øke kolonnebredden for de kolonnene som var for smale. En annen ting vi vil gjøre nå er å binde sammen kjønn med vekt-variablene for 1998 og 1999. Det gjør vi i COLUMN-setningen. Ved å sette de to vekt-variablene i parentes og sette ett komma mellom dem og kjønn blir de knyttet sammen. Kommaet brukes altså her til å knytte sammen kolonner, ikke til å skille mellom dimensjoner slik som i TABULATE. Vi velger dessuten kun de med gyldige kjønnskoder.

```

PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE kjonn IN('1','2');
  COLUMN naering kjonn, (vekt1998-vekt1999);
  DEFINE naering / GROUP WIDTH=34 SPACING=0;
  DEFINE kjonn / ACROSS WIDTH=5 ' ';
  DEFINE vekt1998 / ANALYSIS SUM WIDTH=8 '1998' F=8.;
  DEFINE vekt1999 / ANALYSIS SUM WIDTH=8 '1999' F=8.;
  FORMAT kjonn $kjonn.
         naering $nar.;
  TITLE 'Rapport 5. Sysselsatte gruppert etter næring og kjønn' ;
RUN;

```

Tabellen ser nå ganske brukbar ut:

Rapport 5. Sysselsatte gruppert etter næring og kjønn				
Næring	Kvinner		Menn	
	1998	1999	1998	1999
Bergverksdrift	56352	53100	176330	175780
Bygge- og anleggsvirksomhet	.	.	42240	48635
Fiske, fangst og fiskeoppdrett	56352	58100	42240	45635
Industri	690594	697574	702129	737783
Jordbruk, jakt og viltstell	.	.	47363	48323
Kraft- og vannforsyning	56352	56100	34241	38811
Skogbruk	34241	38811	.	.
Utv.av råolje og naturgass,inkl.tj	.	.	34241	38811

En ting vi savner ved denne tabellen er en totallinje. Den legger vi inn ved å bruke RBREAK-setningen. Med den sørger vi for at det lages en totallinje.

```
PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE kjønn IN('1','2');
  COLUMN næring kjønn, (vekt1998-vekt1999);
  DEFINE næring / GROUP WIDTH=34 SPACING=0;
  DEFINE kjønn / ACROSS WIDTH=5 ' ';
  DEFINE vekt1998 / ANALYSIS SUM WIDTH=8 '1998' F=8.;
  DEFINE vekt1999 / ANALYSIS SUM WIDTH=8 '1999' F=8.;
  RBREAK BEFORE / SUMMARIZE SKIP;
  FORMAT kjønn $kjønn.
           næring $nar.;
  TITLE 'Rapport 6. Sysselsatte gruppert etter næring og kjønn' ;
RUN;
```

Selve summeringen ber vi om med SUMMARIZE. Vi får en blank linje etter sumlinjen med SKIP. Totaltekst får vi derimot ikke uten å jobbe mer for det, mer om det senere:

Rapport 6. Sysselsatte gruppert etter næring og kjønn				
Næring	Kvinner		Menn	
	1998	1999	1998	1999
	893891	903685	1078784	1133778
Bergverksdrift	56352	53100	176330	175780
Bygge- og anleggsvirksomhet	.	.	42240	48635
Fiske, fangst og fiskeoppdrett	56352	58100	42240	45635
Industri	690594	697574	702129	737783
Jordbruk, jakt og viltstell	.	.	47363	48323
Kraft- og vannforsyning	56352	56100	34241	38811
Skogbruk	34241	38811	.	.
Utv.av råolje og naturgass,inkl.tj	.	.	34241	38811

Hvordan lage mer avanserte tabeller i REPORT

Beregne nye kolonner

Det er ikke alltid vi kan lage tabellen vår helt ferdig bare ved hjelp av de variablene som finnes på SAS datasettet. Da kan vi i en del tilfeller bruke beregnede variable i REPORT. Disse må først defineres i COLUMN-setningen og da bruker vi navn som ikke finnes på SAS datasettet fra før. Beregnede variable kan bare benytte andre variable som er definert tidligere i COLUMN-setningen enn hjelpevariabelen selv. Under følger et eksempel der vi summerer to kolonner.

```

PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE kjonn IN('1','2');
  COLUMN naering kjonn,vekt1999 beggekjonn;
  DEFINE naering / GROUP WIDTH=34 SPACING=0;
  DEFINE kjonn / ACROSS WIDTH=5 ' ';
  DEFINE vekt1999 / ANALYSIS SUM WIDTH=8 ' ' F=8.;
  DEFINE beggekjonn/ COMPUTED F=8. 'Begge kjønn';
  COMPUTE beggekjonn;
    beggekjonn = _C2_ + _C3_;
  ENDCOMP;
  RBREAK BEFORE / SUMMARIZE SKIP;
  FORMAT kjonn $kjonn.
         naering $nar.;
  TITLE 'Rapport 7. Sysselsatte gruppert etter næring og kjønn. 1999' ;
RUN;

```

Hver kolonne vi definerer får et eget internt navn i REPORT. De heter `_C1_`, `_C2_` osv. I tabellen under lager vi sum-kolonnen som en beregnet variabel. Først nevner vi den etter kjønn i COLUMN-setningen. Deretter definerer vi den som beregnet i DEFINE-setningen. Til slutt utfører vi beregningen i en COMPUTE-blokk. Vi ser at vi finner summen av begge kjønn ved å summere kolonne `_C2_` og `_C3_`. Vi må være sikre på at det er disse kolonnene som skal summeres og passe på å sjekke summeringen om vi legger til flere variable i COLUMN-setningen. Legger vi til noe før kjønn, vil kolonnene med kjønn forskyve seg og vi må endre hvilke kolonner som inngår i beregningen.



Rapport 7. Sysselsatte gruppert etter næring og kjønn. 1999

Næring	Kvinner	Menn	Begge kjønn
	903685	1133778	2037463
Bergverksdrift	53100	175780	228880
Bygge- og anleggsvirksomhet	.	48635	.
Fiske, fangst og fiskeoppdrett	58100	45635	103735
Industri	697574	737783	1435357
Jordbruk, jakt og viltstell	.	48323	.
Kraft- og vannforsyning	56100	38811	94911
Skogbruk	38811	.	.
Utv.av råolje og naturgass,inkl.tj	.	38811	.

Nå utvider vi tabellen med en andelskolonne. Som istad må den først defineres i COLUMN-setningen, så i DEFINE-setningen og til slutt gjøres selve beregningen med COMPUTE.

```

PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE kjonn IN('1','2');
  COLUMN naering kjonn,vekt1999 beggekjonn andelkvinner;
  DEFINE naering      / GROUP    WIDTH=34 SPACING=0;
  DEFINE kjonn        / ACROSS   WIDTH=5  '  ';
  DEFINE vekt1999     / ANALYSIS SUM WIDTH=8  '  ' F=8.;
  DEFINE beggekjonn   / COMPUTED F=8. 'Begge kjønn';
  DEFINE andelkvinner/ COMPUTED F=7.2 'Andel kvinner';
  COMPUTE beggekjonn;
    beggekjonn = _C2_ + _C3_;
  ENDCOMP;
  COMPUTE andelkvinner;
    andelkvinner= _C2_ /beggekjonn;
  ENDCOMP;
  RBREAK BEFORE / SUMMARIZE SKIP;
  FORMAT kjonn $kjonn.
         naering $nar.;
  TITLE 'Rapport 8. Sysselsatte gruppert etter næring og kjønn. 1999' ;
RUN;

```

Hvis vi bytter rekkefølgen på kjønnskolonnenne må vi også passe på å endre andelsberegningen.

Rapport 8. Sysselsatte gruppert etter næring og kjønn. 1999

Næring	Kvinner	Menn	Begge kjønn	Andel kvinner
	903685	1133778	2037463	0.44
Bergverksdrift	53100	175780	228880	0.23
Bygge- og anleggsvirksomhet	.	48635	.	.
Fiske, fangst og fiskeoppdrett	58100	45635	103735	0.56
Industri	697574	737783	1435357	0.49
Jordbruk, jakt og viltstell	.	48323	.	.
Kraft- og vannforsyning	56100	38811	94911	0.59
Skogbruk	38811	.	.	.
Utv.av råolje og naturgass,inkl.tj	.	38811	.	.



I våre publiseringer ønsker vi å ha sumtall øverst og til venstre. Vi får en sum-linje ved å bruke opsjonen BEFORE i RBREAK-setningen. For å kolonnen med begge kjønn foran fordeling etter kjønn må vi derimot trikse litt. Siden de variablene vi skal bruke i en beregning må være definert før den beregnede variabelen, introduserer vi en beregnet variabel for hvert kjønn. Disse definerer vi etter begge kjønn. Vi vil nå få dobbelt opp med kjønnskolonner, og fjerner derfor de som står feilplassert ved hjelp av NOPRINT-opsjonen i DEFINE-setningen:


```

PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE kjonn IN('1','2');
  COLUMN naering kjonn,vekt1999 beggekjonn menn kvinner andelkvinner;
  DEFINE naering / GROUP WIDTH=34 SPACING=0 ORDER=internal;
  DEFINE kjonn / ACROSS NOPRINT WIDTH=5 ' ';
  DEFINE vekt1999 / ANALYSIS SUM WIDTH=8 ' ' F=8.;
  DEFINE beggekjonn / COMPUTED F=8. 'Begge kjønn';
  DEFINE menn / COMPUTED 'Menn';
  DEFINE kvinner / COMPUTED 'Kvinner';
  DEFINE andelkvinner/ COMPUTED F=7.2 'Andel kvinner';
  COMPUTE beggekjonn;
    beggekjonn = _C2_ + _C3_;
  ENDCOMP;
  COMPUTE menn;
    menn = _C3_ ;
  ENDCOMP;
  COMPUTE kvinner;
    kvinner = _C2_ ;
  ENDCOMP;
  COMPUTE andelkvinner;
    andelkvinner= kvinner /beggekjonn;
  ENDCOMP;
  RBREAK BEFORE / SUMMARIZE SKIP;
  FORMAT kjonn $kjonn.
         naering $nar.;
  TITLE 'Rapport 9. Sysselsatte gruppert etter næring og kjønn. 1999' ;
RUN;

```

Vi får nå ønsket rekkefølge for kolonnene våre også:

Rapport 9. Sysselsatte gruppert etter næring og kjønn. 1999				
Næring	Begge kjønn	Menn	Kvinner	Andel kvinner
	2037463	1133778	903685	0.44
Jordbruk, jakt og viltstell	.	48323	.	.
Skogbruk	.	.	38811	.
Fiske, fangst og fiskeoppdrett	103735	45635	58100	0.56
Bergverksdrift	228880	175780	53100	0.23
Utv.av råolje og naturgass,inkl.tj	.	38811	.	.
Industri	1435357	737783	697574	0.49
Kraft- og vannforsyning	94911	38811	56100	0.59
Bygge- og anleggsvirksomhet	.	48635	.	.

Flere nivåer i forspalten

Når vi skal lage en tabell med flere nivåer i forspalten vil hvert nivå komme i en egen kolonne. I REPORT kalles skillene mellom nivåene for grupperingsvariable for brudd og de defineres med RBREAK-og BREAK-setninger. RBREAK gir brudd på totalnivået (=totalsummer), mens BREAK gir brudd for underliggende nivåer. Det er ingen automatisk summering over nivåene, det må vi angi med SUMMARIZE i RBREAK/BREAK-setninger. Under følger et eksempel på en tabell med to nivåer i forspalten. Vi har valgt ut noen næring og gruppert dem som tidligere:

```
PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE kjonn IN('1','2') AND '15' <= naering <='41';
  COLUMN naering kjonn vekt1998-vekt1999;
  DEFINE naering      / GROUP   WIDTH=34 SPACING=0 ORDER=internal;
  DEFINE kjonn        / GROUP   WIDTH=7  ' ' ;
  DEFINE vekt1998     / ANALYSIS SUM WIDTH=8 '1998' F=8.;
  DEFINE vekt1999     / ANALYSIS SUM WIDTH=8 '1999' F=8.;
  RBREAK BEFORE / SUMMARIZE SKIP;
  FORMAT kjonn $kjonn.
           naering $nar.;
  TITLE 'Rapport 10. Sysselsatte gruppert etter næring og kjønn. ' ;
RUN;
```

Vi ser at det blir en kolonne for hvert nivå. Dessuten får vi ingen summer for næringsgruppene:

Rapport 10. Sysselsatte gruppert etter næring og kjønn.			
Næring		1998	1999
		1483316	1530268
Industri	Kvinner	690594	697574
	Menn	702129	737783
Kraft- og vannforsyning	Kvinner	56352	56100
	Menn	34241	38811

For å få delsummer for næringsgruppene bruker vi altså BREAK-setningen. Med den kan vi sørge for å få inn delsummene (SUMMARIZE) og også legge inn blanke linjer (SKIP).

```
PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE kjonn IN('1','2') AND '15' <= naering <='41';
  COLUMN naering kjonn vekt1998-vekt1999;
  DEFINE naering      / GROUP   WIDTH=34 SPACING=0 ORDER=internal;
  DEFINE kjonn        / GROUP   WIDTH=7  ' ' ;
  DEFINE vekt1998     / ANALYSIS SUM WIDTH=8 '1998' F=8.;
  DEFINE vekt1999     / ANALYSIS SUM WIDTH=8 '1999' F=8.;
  RBREAK BEFORE / SUMMARIZE SKIP;
  BREAK  BEFORE naering/ SUMMARIZE ;
  FORMAT kjonn $kjonn.
           naering $nar.;
  TITLE 'Rapport 11. Sysselsatte gruppert etter næring og kjønn. ' ;
RUN;
```

En ulempe her er at næringsgruppene repeteres i forspalten:

Rapport 11. Sysselsatte gruppert etter næring og kjønn.			
Næring		1998	1999

		1483316	1530268
Industri		1392723	1435357
Industri	Kvinner	690594	697574
	Menn	702129	737783
Kraft- og vannforsyning		90593	94911
Kraft- og vannforsyning	Kvinner	56352	56100
	Menn	34241	38811

Vi kan legge inn blanke linjer også før et brudd i hierarkiene. Det kan gjøres slik:

```
PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE kjonn IN('1','2') AND '15' <= naering <='41';
  COLUMN naering kjonn vekt1998-vekt1999;
  DEFINE naering / GROUP WIDTH=34 SPACING=0 ORDER=internal;
  DEFINE kjonn / GROUP WIDTH=7 ' ' ;
  DEFINE vekt1998 / ANALYSIS SUM WIDTH=8 '1998' F=8.;
  DEFINE vekt1999 / ANALYSIS SUM WIDTH=8 '1999' F=8.;
  RBREAK BEFORE / SUMMARIZE SKIP;
  BREAK BEFORE naering/ SUMMARIZE ;
  BREAK AFTER naering/ SKIP ;
  FORMAT kjonn $kjonn.
         naering $nar.;
  TITLE 'Rapport 12. Sysselsatte gruppert etter næring og kjønn. ' ;
RUN;
```

Vi lager et brudd før næring for å få summene først. Så lager vi et brudd etter næring for å legge inn blank linje etter hver næring. Disse linjene legges inn etter at all utskrift for en næring er gjort, også næringens underliggende nivåer skrives ut før den blanke linjen.

Rapport 12. Sysselsatte gruppert etter næring og kjønn.			
Næring		1998	1999

		1483316	1530268
Industri		1392723	1435357
Industri	Kvinner	690594	697574
	Menn	702129	737783
Kraft- og vannforsyning		90593	94911
Kraft- og vannforsyning	Kvinner	56352	56100
	Menn	34241	38811

Lage en sammensatt forspalte ut fra to grupperinger

Som vi så i eksemplene over blir ikke forspalten helt som ønskelig ved å definere flere grupperingsnivåer i forspalten. Vi så at næringsteksten ble gjentatt flere ganger og at kjønn havnet i en egen kolonne. Det vi vanligvis ønsker av en forspalte som har flere grupperingsnivåer er at teksten på det høyeste nivået (her: næring) skal komme en gang, på den linjen sumtallene for næringene havner i. Tilsvarende for underliggende nivå (her: kjønn). Teksten skal dessuten rykkes litt inn i forhold til tekst på overliggende nivå. Det skal også være blanke linjer for å gjøre tabellen mer leselig, de er allerede lagt inn på riktige steder. Vi har tidligere sett at vi kan beregne nye kolonner ut fra eksisterende. Det vanlige er da å lage sum- prosent- og rate-kolonner. For å lage en sammensatt forspalte vil vi nå bruke en hjelpevariabel SAS selv bruker til å skille de forskjellige nivåene i forspalten. Denne hjelpevariabelen heter `_BREAK_`. Det kan være greit å se hva den inneholder på de enkelte linjene i tabellen og det får vi til ved å gjøre programmet slik:



```

PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
WHERE kjonn IN('1','2') AND '15' <= naering <='41';
COLUMN naering kjonn brudd vekt1998-vekt1999;
DEFINE naering / GROUP WIDTH=23 SPACING=0 ORDER=internal;
DEFINE kjonn / GROUP WIDTH=7 ' ' ;
DEFINE vekt1998 / ANALYSIS SUM WIDTH=8 '1998' F=8.;
DEFINE vekt1999 / ANALYSIS SUM WIDTH=8 '1999' F=8.;
DEFINE brudd / COMPUTED '_BREAK_' WIDTH=9 ;
RBREAK BEFORE / SUMMARIZE SKIP;
BREAK BEFORE naering/ SUMMARIZE ;
BREAK AFTER naering/ SKIP ;
COMPUTE brudd /CHARACTER LENGTH=9;
 brudd = _BREAK_;
ENDCOMP;
FORMAT naering $nar. ;
TITLE 'Rapport 13. Sysselsatte gruppert etter naering og kjonn. ' ;
RUN;

```

Vi ser at kolonnen `_BREAK_` får forskjellige verdier for de forskjellige nivåene i tabellen. Dette vil vi utnytte når vi skal lage en ny, beregnet kolonne med den sammensatte forspalten. Vi ser at `_BREAK_` får verdien `_RBREAK_` på totalnivået, Naering for næringsnivået og blank for kjønnsnivået.

Rapport 13. Sysselsatte gruppert etter naering og kjonn.				
Naering		<code>_BREAK_</code>	1998	1999
		<code>_RBREAK_</code>	1483316	1530268
Industri		Naering	1392723	1435357
Industri	1		702129	737783
	2		690594	697574
Kraft- og vannforsyning		Naering	90593	94911
Kraft- og vannforsyning	1		34241	38811
	2		56352	56100

Vi lager den nye, sammensatte kolonnen ved hjelp av en `SELECT`-setning. Det lages en betingelse for hvert nivå, og vi benytter oss av `_BREAK_`-variabelen ved testingen. Avhengig av hvilke tester som slår til vil vi legge tekster for de riktige nivåene på riktig sted. Tekster for naering og kjonn henter vi ved hjelp av `PUT`-funksjonen. Denne henter teksten til et format ut fra den koden som sendes inn til funksjonen. For å få innrykket teksten til kjonn, legger vi to blanke først i den beregnede variabelen. En viktig ting å legge merke til i programmet under er at vi formaterer naering både med en `FORMAT`-setning når tabellen bygges opp og ved en `PUT`-setning når vi lager den beregnede kolonnen. Det må vi gjøre siden vi har laget et format som grupperer verdier. Hvis vi ikke hadde brukt `FORMAT`-setningen ville ikke naeringene blitt gruppert i tabellen. Vi kan nå fjerne kolonnen med brudd-indikatoren i tabellen vår. Vi kan også undertrykke de opprinnelige forspaltekolonnene. Det gjør vi med opsjonen `NOPRINT` i `DEFINE`-setningen:

```

PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  WHERE kjonn IN('1','2') AND '15' <= naering <='41';
  COLUMN naering kjonn forsp vekt1998-vekt1999;
  DEFINE naering      / GROUP NOPRINT WIDTH=23 SPACING=0 ORDER=internal;
  DEFINE kjonn        / GROUP NOPRINT WIDTH=7 ' ' ;
  DEFINE vekt1998     / ANALYSIS SUM WIDTH=8 '1998' F=8.;
  DEFINE vekt1999     / ANALYSIS SUM WIDTH=8 '1999' F=8.;
  DEFINE forsp        / COMPUTED 'Næring og kjønn' WIDTH=23 SPACING=0;
  RBREAK BEFORE / SUMMARIZE SKIP;
  BREAK BEFORE naering/ SUMMARIZE ;
  BREAK AFTER naering/ SKIP ;
  COMPUTE forsp /CHARACTER LENGTH=35;
  SELECT;
    WHEN (_BREAK_ = '_RBREAK_') forsp= 'I alt';
    WHEN (UPCASE(_BREAK_) = 'NAERING') forsp= PUT(naering,$nar.);
    WHEN (_BREAK_ = '') forsp= ' ' || PUT(kjonn,$kjonn.);
    OTHERWISE ;
  END;
  ENDCOMP;
  FORMAT naering $nar.;
  TITLE 'Rapport 14. Sysselsatte gruppert etter næring og kjønn. ' ;
RUN;

```

Tabellen har nå fått en akseptabel forspalte:

Rapport 14. Sysselsatte gruppert etter næring og kjønn.

Næring og kjønn	1998	1999
I alt	1483316	1530268
Industri	1392723	1435357
Kvinner	690594	697574
Menn	702129	737783
Kraft- og vannforsyning	90593	94911
Kvinner	56352	56100
Menn	34241	38811

Det vi kan savne i tabellen over er en kjønnsfordeling for totalnivået. En slik fordeling er enkelt å lage i TABULATE, men verre å få til i REPORT. En mulighet i REPORT for å ordne dette er å aggregere tallene på forhånd med PROC MEANS, for så å bruke REPORT til å skrive dem ut. Denne teknikken krever at vi ikke har noen data uten koder for grupperingsvariablene, altså ingen manglende verdier for næring eller kjønn. Da kan resultatet fort bli feil. I og med at vi aggregerer alle tall på forhånd vil vi ikke kunne gjøre nytte av _BREAK_-variabelen for å skille nivåene fra hverandre. Da må vi i stedet bruke informasjonen vi kan hente ut fra de to kolonnene som utgjør forspalten (her: næring og kjønn). Vi vil se at vi er på det laveste nivået (kjønn) når det finnes en tekst i dens kolonne. Dette utnytter vi ved den første testen. Videre vil det være slik at det vil være tekst på det overliggende nivå (næring) når denne kolonnen har en tekst. Dette gjelder bare når vi har utført testen på det laveste nivået først. Når disse to testene er utført er det bare en mulighet igjen og det er at vi befinner oss på totalinjen. Programmet vil se slik ut, rekkefølgen på testene er viktig:

```

PROC MEANS DATA=aku NOPRINT ;
  WHERE kjonn IN('1','2') AND '15' <= naering <='41';
  CLASS naering kjonn;
  VAR vekt1998 vekt1999;
  OUTPUT OUT=aku_a SUM=;
RUN;

PROC REPORT DATA=aku_a HEADLINE HEADSKIP NOWINDOWS MISSING;
  COLUMN naering kjonn forsp vekt1998-vekt1999;
  DEFINE naering / GROUP NOPRINT WIDTH=23 SPACING=0 ORDER=internal;
  DEFINE kjonn / GROUP NOPRINT WIDTH=7 ' ' ;
  DEFINE vekt1998 / ANALYSIS SUM WIDTH=8 '1998' F=8.;
  DEFINE vekt1999 / ANALYSIS SUM WIDTH=8 '1999' F=8.;
  DEFINE forsp / COMPUTED 'Næring og kjønn' WIDTH=23 SPACING=0;
  BREAK AFTER naering/ SKIP ;
  COMPUTE forsp /CHARACTER LENGTH=35;
  SELECT;
  WHEN (kjonn NE '') forsp= ' ' || PUT(kjonn,$kjonn.);
  WHEN (naering NE '') forsp= PUT(naering,$nar.);
  OTHERWISE forsp= 'I alt';
  END;
  ENDCOMP;
  FORMAT naering $nar.;
  TITLE 'Rapport 15. Sysselsatte gruppert etter næring og kjønn. ' ;
RUN;

```

Nå ser vi at vi har fått med delsummer for kjønn under totalsummen.

Rapport 15. Sysselsatte gruppert etter næring og kjønn.		
Næring og kjønn	1998	1999
I alt	1483316	1530268
Menn	736370	776594
Kvinner	746946	753674
Industri	1392723	1435357
Menn	702129	737783
Kvinner	690594	697574
Kraft- og vannforsyning	90593	94911
Menn	34241	38811
Kvinner	56352	56100

Vi må være klar over at underliggende nivå bare blir skrevet ut når det finnes data for det. Det betyr i vårt tilfelle for eksempel at det ikke ville bli skrevet ut noen linje for Kvinner hvis det var en næring der kvinner ikke var representert.

Utfyllende om syntaksen til REPORT

```

PROC REPORT <options>;
WHERE where-uttrykk ;
BY <DESCENDING> variabel-1 ... <<DESCENDING> variabel-n> <NOTSORTED>;
COLUMN kolonnespesifikasjoner ;
DEFINE report-variabel </ <options> <'overskrift-1' <... 'overskrift-n'>>;
BREAK brudd-variabel </brudd-options> ;
RBREAK bruddplassering </brudd-options> ;
COMPUTE plassering <brudd-variabel> | <_PAGE_ <LEFT | CENTER | RIGHT>>
    </ STYLE=[stil-spesifikasjoner] ;
    LINE <peker> variabel <format> | <n*> 'tekst' ;
ENDCOMP;
COMPUTE report-variabel </ CHARACTER <LENGTH=lengde>>;
    CALL DEFINE (_COL_, 'attributtnavn' | _ROW_, attributtverdi);
ENDCOMP;
FORMAT variabel utskriftsformat. <variable utskriftsformat.>;
FREQ variabel;
WEIGHT variabel;

```

Options til PROC REPORT-setningen:

BOX

Angir at rapporten og de enkelte celler skal rammes inn med de tegnene som er definert med system option FORMCHAR. Kan ikke brukes sammen med WRAP-option eller med FLOW-option i DEFINE-setningen.

CENTER

Angir om rapporten skal sentreres (CENTER) eller venstrejusteres (NOCENTER).

COLWIDTH=kolonnebredde

Angir standard bredde for kolonnene i rapporten.

COMMAND

Angir at kommandolinje skal vises i REPORT-vinduene når vi bruker dem.

COMPLETECOLS

Sørger for å skrive ut alle kolonner som er mulige, selv om de ikke vil inneholde tall. Se også PRELOADFMT-option til DEFINE-setningen.

COMPLETEROWS

Sørger for å skrive ut alle rader som er mulige, selv om de ikke vil inneholde tall. Se også PRELOADFMT-option til DEFINE-setningen.

CONTENTS='tekst'

Angir tekst til html innholdsfil (laget med CONTENTS= i ODS-setningen) for rapporten.

DATA= SAS datasett

Her oppgis navnet på SAS datasettet vi ønsker å lage rapporten fra.

EXCLNPWGT

Ekskluderer observasjoner der vektvariabelens verdi (definert i WEIGHT-setningen) er 0 eller mindre. Uten denne option regnes negative verdier for vektvariablene som 0 og observasjonene inngår i optelling av antall observasjoner (N).

FORMCHAR

Brukes for å angi tegn for innramming av tabellen. Tegnenes rekkefølge er de samme som i TABULATE: 1. er vertikal strek (| er standard), 2. er horisontal strek (-), 3. er øvre venstre hjørne (-), 4. er midten øverst (-), 5. er øvre høyre hjørne (-), 6. er midten til venstre (|), 7. er midt i tabellen (+), 8. er midten til høyre (|), 9. er nedre venstre hjørne (-), 10. er midten nederst (-) og 11. er nedre høyre hjørne (-). I tillegg brukes 13 til å angi tegn for dobbel understreking av sumlinjer (=). Vi kan f.eks. endre tegn slik: FC(7,13)='-_'

HEADLINE

Legger inn en strek under kolonneoverskriftene.

HEADSKIP

Legger inn en blank linje under kolonneoverskriftene.

HELP=libref.katalog

Angir en katalog for hjelpeskjermbilder.

LIST

Lister ut SAS-programmet som er kjørt til SASlog. Det programmet SAS kjører avviker noe fra det vi skriver uten at det skal ha noen betydning.

LS=linjestørrelse

Angir linjestørrelsen for rapporten.

MISSING

Sørger for at MISSING behandles som gyldig verdi for grupperingsvariable (definert med GROUP, ORDER eller ACROSS i DEFINE-setningen). Påvirker ingen statistiske beregninger.

NAMED

Skriver kolonneoverskriften ut foran hvert tall i tabellen.

NOEXEC

Gjør at rapporten ikke lages, men programmet syntakssjekkes.

NOHEADER

Undertrykker kolonneoverskrifter.

OUT=

Navn på SAS datasett rapporten skal skrives ut til. SAS datasettet vil inneholde en variabel for hver kolonne i rapporten og en observasjon for hver linje. I tillegg vil det lages en variabel kalt `_BREAK_` når vi bruker `BREAK-` eller `RBREAK-` setningene. Denne inneholder navn på bruddvariabelen for hver enkelt observasjon. Grupperingskolonner i hodet (definert som `ACROSS` i `DEFINE`-setningen) vil få kolonnennummeret som variabelnavn etter mønster `_Cn_`, der `n` er kolonnennummeret. Kolonneoverskrifter vil bli brukt som utfyllende tekst (`LABEL`). Vi kan ikke bruke `OUT=` sammen med `BY`-setningen i `REPORT`.

OUTREPT=libref.katalog.programnavn

brukes for å lagre rapporten i en SAS-katalog. Dette gjør vi vanligvis ikke, vi lagrer programmet i programfiler isteden.

PANELS=spalter

Angir antall spalter rapporten skal skrives ut i på hver side.

PROFILE=libref.katalog

Navngir en SAS-katalog som inneholder en standardprofil for `REPORT`.

PROMPT

Angir at vi får opp et dialogbildet når vi lager rapporten

PS=sidestørrelse

Angir sidestørrelsen for rapporten.

PSPACE=n

Antall posisjoner mellom hver spalte. Brukes sammen med `PANELS`. Standard er 4.

REPORT=libref.katalog.programnavn

Navngir et program som inneholder en standardrapport

SHOWALL

Sørger for å skrive alle kolonner og verdier, selv om de egentlig er undertrykket med `NOPRINT-` eller `NOZERO-` option i `DEFINE`-setningen.

SPACING=n

Antall posisjoner mellom kolonnene i rapporten. Standard er 2.

SPLIT='tegn'

Angir et tegn for linjebrydd i kolonneoverskriftene. Tegnet kommer ikke med i utskriften. Standard er linjebrydd ved `/`.

STYLE<(rapportdel)> = <stilnavn> <(stilattributter)>

Brukes for å endre utseende på forskjellige deler av rapporten for html-filer. Rapportdelene er:

<code>CALLDEF</code>	Alle celler identifisert ved <code>CALL DEFINE</code> -setninger
<code>COLUMN</code>	Cellene i kolonnene
<code>HEADER</code>	Kolonneoverskrifter
<code>LINES</code>	Alle linjer definert med <code>LINE</code> -setninger
<code>REPORT</code>	Generelt for hele rapporten

THREADS

Åpner for parallellprosessering av dataene over flere disketter samtidig. Dette vil senke prosesseringstiden.

VARDEF=divisor

Angir divisor som brukes til å regne ut varianser. Gyldige verdier er:

DF	N-1 skal brukes som divisor (standard)
N	N skal brukes som divisor
WDF	Summen av vektene - 1 skal brukes som divisor
WEIGHT	Summen av vektene skal brukes som divisor

WINDOWS

WINDOWS gjør at rapporten havner i et eget rapportvindu, mens NOWINDOWS sørger for at rapporten havner i OUTPUT-vinduet (SASlist).

WRAP

Lister ut alle verdiene for en logisk linje i rapporten på linjer etter hverandre istedenfor å liste ut kolonnene det er plass til på linjen i puljer.

Andre options: QMARKERS, QMETHOD og QNTLDEF.

SAS-setninger til REPORT:

WHERE

Selekterer hvilke observasjoner det skal lages rapport for. Bare de som tilfredsstillter WHERE-betingelsen tas med.

BY

Det vil lagees en rapport for hver verdi av BY-variablene. Datasettet må være sortert etter BY-variablene på forhånd (med mindre vi bruker sorteringsoption NOTSORTED).

COLUMN

Angir hvilke variable som skal brukes i rapporten. Alle variable som skal brukes på en eller annen måte må angis her. I kolonnedefinisjonene kan vi også angi disse statistikkene; CSS, CV, MAX, MEAN, MIN, N, NMISS, PCTN, PCTSUM, PROBT, RANGE, STD, STDERR, SUM, SUMWGT, T, USS, og VAR, for nærmere beskrivelse, se side 74. N kan brukes på alle typer variable, når denne knyttes mot analysevariable telles antall observasjoner som ikke er MISSING for denne variabelen, for grupperingsvariable fordeles antall observasjoner i SAS datasettet etter grupperingsvariabelens verdier og MISSING regnes som en egen verdi. Denne kommer med som en egen kategori hvis MISSING-option er med i PROC REPORT-setningen. De andre statistikkene må knyttes til analysevariable. Vi kan velge blant disse kvantile statistikkene: MEDIAN, P1, P10, P5, P90, P95, P99, Q1, Q3 og QRANGE. I tillegg har vi PCTN og PCTSUM som regner ut prosent av henholdsvis antall observasjoner og summen av en tilordnet analysevariabel. Prosentene vises som andelstall, ikke som prosent. Ved bruk av vektor vil PCTSUM veies, PCTN derimot veies ikke.

Vi bruker komma for å knytte variable og statistikker sammen. Vi kan f. eks. ha en grupperingsvariabel i hodet (defineres med ACROSS i DEFINE-setningen) som knyttes sammen med en analysevariabel (defineres med ANALYSIS i DEFINE-setningen) som igjen kan knyttes sammen med en eller flere statistikker (f. eks. MEAN og SUM). Vi kan ikke knytte statistikker til mer enn et nivå i disse definisjonene.

Vi kan også skrive overskrifter i kolonnedefinisjonen. Det gir oss blant annet muligheten til å få en overskrift over flere kolonner (variable) og heltrukne linjer over flere kolonner.

Syntaks for overskrifter:

('overskrift-1' <... 'Overskrift-n'> kolonnespesifikasjoner)

Denne syntaksen brukes for å få felles overskrifter over kolonner som skal ha dette, men som ikke vil få det automatisk.

Hvis første og siste tegn i overskriften er -, =, _, ., *, or +, brukes det tegnet til å fylle ut kolonneoverskriften på hver side av teksten. Tilsvarende hvis vi setter teksten inne i <> vil det

fylles ut med < foran og > etter teksten. Dette kan vi bruke til å få streket under tekst i kolonneoverskriftene og også til å få en streklinje etter overskriften.

Ønsker vi å bruke en variabel eller statistikk på flere måter i en og samme rapport, kan vi gi den et alias:

```
variabel = alias
statistikk = alias
```

Samme variabel eller statistikk kan gis flere alias, men hver variabel og alias kan bare brukes i en DEFINE-setning hver.

DEFINE

Brukes til å definere kolonnene rapporten skal bestå av. Vi trenger en DEFINE-setning pr. del av rapporten, vanligvis tilsvarer hver del én kolonne. Hver variabel må angis med en av disse options for bruksmåte:

ACROSS	Variabelen grupperes i kolonner. Det lages en kolonne for hver verdi variabelen har, eller en kolonne for hver formaterte verdi (kodetekst). Når vi bruker formater plasseres kolonnene stigende etter formatenes verdi. Ellers plasseres kolonnene stigende etter variabelens verdi. Dette kan overstyres med ORDER-option og/eller DESCENDING-option i DEFINE-setningen. ACROSS tilsvarer å bruke en CLASS-variabel i hodet i TABULATE.
ANALYSIS	Variabelen må være numerisk og skal brukes til en statistikk, f. eks SUM el. MEAN. Hvis vi ikke angir noen statistikk i DEFINE-setningen, brukes SUM.
COMPUTED	Definerer en hjelpevariabel som ikke finnes på SAS datasettet rapporten lages fra, men som skal beregnes i rapportprosedyren. Variabelens plassering angis i COLUMN-setningen og den defineres i DEFINE-setningen, mens den gis verdi i en COMPUTE-blokk, se COMPUTE-setningen.
DISPLAY	Variable definert med DISPLAY, vil bli listet ut for hver observasjon i SAS-datasettet, disse får altså ikke gruppert like verdier.
GROUP	Variabelen grupperes slik at like variabelverdier grupperes til hver sin linje i rapporten. Når vi bruker formater plasseres kolonnene stigende etter formatenes verdi. Ellers plasseres kolonnene stigende etter variabelens verdi. Dette kan overstyres med ORDER-option og/eller DESCENDING-option i DEFINE-setningen. GROUP tilsvarer å bruke en grupperingsvariabel (definert i en CLASS-setning) i forspalten i TABULATE.
ORDER	Variable definert med ORDER listes ut i sortert rekkefølge for hver observasjon i SAS datasettet. Sorteringsrekkefølgen er etter formaterte verdier når vi knytter et format til variabelen, ellers sorteres de etter variabelens verdi. Dette kan overstyres med ORDER-option og/eller DESCENDING-option i DEFINE-setningen.

Vi trenger ikke å angi bruksmåte når vi definerer en statistikkberegning i DEFINE-setningen, men kan gjerne gjøre det for oversiktens skyld.

Vi kan også angi disse options i DEFINE-setningen:

CENTER	Midtstiller kolonneoverskriften og kolonnens innhold. Tall justeres allikevel slik at de får rett høyremarg
COLOR=	angir kolonneoverskriftens farge. Virker bare når WINDOWS-option er satt i PROC REPORT-setningen. Disse fargene kan brukes: BLACK, BLUE, BROWN, CYAN, GRAY, GREEN, MAGENTA, ORANGE, PINK, RED, WHITE og YELLOW.
DESCENDING	Angir at variabel med bruksmåte ACROSS, DISPLAY eller GROUP skal sorteres synkende

EXCLUSIVE	Ekkluderer variable med kombinasjoner av grupperingsvariable som ikke finnes i formatet som skal være knyttet til variabelen. Krever også at vi bruker PRELOADFMT-option i DEFINE-setningen.
FLOW	Gjør at tekst i kolonnen deles over flere linjer hvis den ikke får plass på avsatt område. Andre kolonnens verdier skrives ut på første linje med tekst. Teksten deles hvis mulig ved delingstegn eller blank.
FORMAT=	Angir hvordan kolonnen skal formateres. Vi kan bruke egendefinerte formater SAS egne formater. Standard er BESTw. for numeriske variable og \$w. for karaktervariable hvor w er standard kolonnebredde. Formater kan også knyttes til variable med en egen FORMAT-setning slik det gjøres i andre prosedyrer.
ID	Gjør at kolonnen og alle kolonner til venstre for ID-variabelen gjentas hvis rapporten må deles fordi den er for bred for siden. Mest aktuelt for variable i forspalten.
kolonneoverskrift	Teksten som skrives i futter vil bli kolonnens overskrift. Teksten brytes hvis den ikke får plass i kolonnebredden. For å sikre riktig tekstbrudd kan vi bruke / eller et annen definert tekstdelingstegn i teksten.
LEFT	Venstrejusterer kolonneoverskriften og kolonnens innhold. Tall justeres allikevel slik at de får rett høyremarg.
MISSING	Sørger for at MISSING behandles som gyldig verdi for variabelen.
NOPRINT	Gjør at kolonnen ikke skrives ut. Overstyres av SHOWALL-option i PROC REPORT-setningen
NOZERO	Undertrykker kolonneutskrift når verdien er 0 eller MISSING. Overstyres av SHOWALL-option i PROC REPORT-setningen
ORDER=	Angir hvilken rekkefølge verdiene til variable med bruksmåte ACROSS, DISPLAY og GROUP skal skrives ut i. Velger vi ORDER=FORMATTED skrives de ut sortert etter teksten i utskriftsformatet som eventuelt brukes istedenfor koden. Dette er standard når vi bruker formater på variabelen. ORDER=FREQ sørger for at den verdi med færrest observasjoner skrives først (med mindre vi også bruker DESCENDING-option også brukes, da kommer den med flest observasjoner kommer først). ORDER=DATA gjør at verdiene beholder den rekkefølgen de har i datasettet, den første forekomsten av verdiene bestemmer rekkefølgen. Standardverdi når vi ikke bruker formater er ORDER=INTERNAL som skriver ut verdiene i sortert rekkefølge. MISSING-verdier vil alltid komme ut først, uansett hvilken rekkefølge vi velger.
PAGE	Gir sideskift før kolonnen skrives ut
PRELOADFMT	Vi bruker PRELOADFMT for å si at vi ønsker å beholde rekkefølgen formatet fikk da det ble lagret. Med denne option og NOTSORTED-option til VALUE-setningen i PROC FORMAT, kan vi sikre at vi får linjene og kolonnene i den riktige rekkefølgen i utskriften. For å få til dette må vi også ta med ORDER=data som option til DEFINE-setningen.
RIGHT	Høyrejusterer kolonneoverskriften og kolonnens innhold.
SPACING=	Angir hvor mange blanke posisjoner som skal skrives ut før kolonnen. Standard er 2.
statistikk	Angir hvilken statistikkberegning vi skal utføre på variable med ANALYSIS som bruksmåte, se COLUMN-setningen for oversikt over statistikker.
STYLE=	Angir stilspesifiseringer til html-utskrift.
WEIGHT=	Angir en vekt-variabel. Hvis vekten er 0 eller negativ settes den til 0 og observasjonen teller med i antallet, men får ingen verdi. Mangler vekten verdi holdes observasjonen utenfor rapporten.
WIDTH=	Angir kolonnebredde
BREAK	BREAK -setningen bruker vi for å definere brudd i forspalten i rapporten. Vi kan ha variable definert med GROUP eller ORDER i DEFINE-setninger som bruddvariable. Vanligvis vil vi lage summer for bruddene, selv om vi kan ha brudd uten å summere også. I BREAK -setningen angir vi

først om bruddlinjen skal plasseres før (BEFORE) eller etter (AFTER) at underliggende nivåer skrives ut. Deretter angir vi hvilken variabel bruddet skal skje ved, en variabel pr. BREAK-setning.

Til slutt kan vi angi følgende options til BREAK-variabelen:

COLOR=farge

Angir en farge for bruddlinjen, alle bokstaver og tall på linjen får den valgte fargen. Virker bare når option WINDOWS er satt i PROC REPORT-setningen. Følgende farger er gyldige: BLACK, BLUE, BROWN, CYAN, GRAY, GREEN, MAGENTA, ORANGE, PINK, RED, WHITE og YELLOW.

DOL

Legger en linje med doble streker før summeringen.

DUL

Legger en linje med doble streker etter summeringen.

OL

Legger en linje med enkle streker før summeringen.

PAGE

Gir sideskift ved brudd. Kan f. eks. brukes til å lage en rapport pr. grupperingsvariabel.

SUMMARIZE

Sørger for at vi får laget summer ved brudd i verdiene i forspalten.

SUPPRESS

Undertrykker verdien til bruddvariabelen (forspaltevariabelen) i utskriften.

SKIP

Gir linjeskift etter brudd. Nyttig for å få inn blanke linjer i rapporten.

STYLE

Til stilsesifikasjoner for html-utskrift.

UL

Legger en linje med enkle streker etter summeringen.

RBREAK

RBREAK-setningen bruker vi for å angi at vi skal ha med totalsummer, enten øverst (BEFORE) eller nederst (AFTER) i rapporten. Vi kan bruke de samme OPTIONS til RBREAK som til BREAK, unntatt PAGE og SUPPRESS, se over.

COMPUTE

Brukes til å utføre beregninger på rapporten. Kan brukes til å avlede nye kolonner fra eksisterende, skifte ut variabelverdier med tekst og utføre forskjellige beregninger. I beregningene i COMPUTE-setningen kan vi ikke bruke variable som er definert lenger til høyre i tabellen enn COMPUTE-variabelen. Vi kan også bruke COMPUTE til å gjøre spesielle ting ved brudd i rapporten, se LINE-setningen, og til å tilpasse utseende på deler av rapporten, se CALL DEFINE. Inne i en COMPUTE-blokk kan vi utføre mange av de vanlige SAS-setninger som vi ellers bruker i DATA-steget:

assignment (tilordning), CALL, DM, DO, END, GO TO, IF-THEN/ELSE, %INCLUDE, kommentar, LENGTH, LINK, RETURN, SELECT og sum, for beskrivelse av disse se kapittelet om DATA-steget. I tillegg til disse setningene, kan vi bruke CALL DEFINE- og LINE-setningene i COMPUTE-blokker. Vi kan også bruke alle SAS funksjoner, makro-variable og kalle opp makroer i COMPUTE-blokker.

Hver COMPUTE-blokk skal avsluttes med en ENDCOMP-setning.

Vi har to varianter av COMPUTE-setningen, en for å gjøre beregninger ved brudd, og en for andre beregninger. Til beregninger ved brudd bruker vi LINE-setningen for å skrive ut egne linjer i forbindelse med bruddet, se LINE-setningen. I COMPUTE-setningen angir vi da om linjene skal plasseres før (BEFORE) eller etter (AFTER) bruddet. Vi kan ha med en brudd-variabel i COMPUTE-setningen, i så fall skrives linjene ut for hvert brudd denne variabelen får. Har vi ikke med noen bruddvariabel, får vi skrevet ut de ønskede linjer enten øverst eller nederst i tabellen. Vi kan også få skrevet ut en tekst først (dvs. etter overskriften) eller sist (dvs. før fotnoter) på hver side ved å bruke `_PAGE_`. Teksten vi skriver ut kan vi da plassere til venstre (LEFT), i midten (CENTER) eller til høyre (RIGHT), eller vi kan bestemme plassering i LINE-setningen der vi angir teksten som skal skrives ut.

For beregninger som gjøres uavhengig av brudd i forspalten, kan vi definere dette i COMPUTE-setningen:

report-variabel angir navnet på den hjelpevariabelen som skal gis verdi. Denne må være nevnt i COLUMN-setningen og definert som COMPUTED i DEFINE-setningen.
 CHARACTER definerer en karaktervariabel
 LENGTH= karaktervariabelens lengde
 STYLE= layout-stil for html-filer.

Når vi henviser til analysevariable for å gjøre beregninger på dem, bruker vi denne formen:

variabel.statistikk

For kolonner laget fra variable definert som ACROSS i DEFINE-setningen, må vi bruke kolonnennummeret som referanse: `_Cn_`, der n er kolonnennummeret. Kolonne `_C1_` er lengst til venstre. Hvis antallet forskjellige verdier for ACROSS-variable endres fra en kjøring til en annen, må vi passe på å sjekke at kolonnennumrene det refereres til er de riktige.

Vi kan også bruke STYLE-option til å angi stilspesifikasjoner til html-utskrift.

CALL DEFINE

Brukes for å angi hvordan en spesiell celle i rapporten skal se ut. CALL DEFINE må stå inne i en COMPUTE-blokk. Vi kan bare endre attributter på kolonner definert som COMPUTED i DEFINE-setningen. COLOR-attributten virker bare når WINDOWS-option er satt i PROC REPORT-setningen. Aktuelle attributter:

color angir en farge innholdet i cellen skal få. Gyldige verdier er BLACK, BLUE, BROWN, CYAN, GRAY, GREEN, MAGENTA, ORANGE, PINK, RED, WHITE og YELLOW.
 format angir formatet innholdet i cellen skal benytte. Dette må stå i fnutter.
 style= stiler for html-utskrift
 url brukes for å lage linker til andre filer (URL=Uniform Resource Locater).
 urlbp brukes for å lage linker til andre filer. Begynnelsen av linken skal være definert i ODS med BASE- eller PATH-option.
 urlp brukes for å lage linker til andre filer. Begynnelsen av linken skal være definert i ODS med PATH-option.

LINE

Angir en linje som skal skrives ut i rapporten. LINE-setningen må plasseres inne i en COMPUTE-blokk og kan bare brukes på spesielle steder. Den må ha en forbindelse til et brudd i forspalten definert med BREAK- eller RBREAK-setninger. LINE-setningen kan ikke brukes i forbindelse med IF, SELECT eller DO-setninger, den utføres uansett sist i COMPUTE-blokken uavhengig av slike tester.

LINE-setningen fungerer omtrent som en PUT-setning i DATA-steget, men har ikke så mange muligheter som denne. Følgende kan angis i LINE-setningen:

peker Angir pekertegn og posisjon for utskrift på linjen. Pekertegn kan være @ eller + som etterfølges av et tall. Tallet etter @ angir posisjon på linja, mens det etter + angir at vi skal forflytte oss et antall posisjoner på linjen før utskrift.
 variabel Hvordan variabelen (egentlig kolonnen) skal refereres avhenger av hvordan den er definert i DEFINE-setningen. Er den definert som GROUP, ORDER, COMPUTED eller DISPLAY, skal referansen være variabelnavnet. Variable definert med ANALYSIS skal refereres med variabelnavn.statistikk (f. eks inntekt.sum). Kolonner som inngår i ACROSS-definisjoner må refereres med kolonnennummeret (f. eks `_C13_`).
 format Formatet kan være egendefinert eller definert av SAS.
 'tekst' Teksten vil komme ut på linjen slik den står i fnuttene. Er teksten blank, kommer det ut en blank linje i rapporten.
 n*'tekst' Teksten i fnutter vil gjentas n ganger på linjen (brukes f. eks. for å få en linje med streker)

ENDCOMP

Avslutter en COMPUTE-blokk.

FORMAT

Vi bruker FORMAT-setningen til å erstatte variabelverdier med mer forståelig *tekst*. Disse utskriftsformatene må være definert på forhand vha. PROC FORMAT. FORMAT-setningen kan også brukes til å *gruppere* variabelverdier, f. eks. etter aldersgrupper eller inntektsgrupper.

WEIGHT

For å veie dataene bruker vi WEIGHT-setningen. Vi må da ha en variabel på datasettet som inneholder vekten (oppblåsingsfaktoren). Vektvariabelen kan gjerne være et desimaltall. Når vektens verdi er mindre enn 0, settes den til 0 og observasjonen telles med i antallet (N). Når verdien er 0 telles observasjonen med i antallet. Dette kan overstyres med REPORT-option EXCLNPWGHTS som ekskluderer disse observasjonene fra tabellen. Er vektvariabelen MISSING, ekskluderes observasjonen fra tabellen.

FREQ

Hvis hver observasjon på datasettet skal telle for mer enn 1, kan vi bruke FREQ-setningen. I FREQ-setningen kan vi da angi variabelen hvis verdi inneholder det antall observasjoner denne observasjonen skal telle som. Dette betyr at N også økes med verdien til FREQ-variabelen.

FREQ trunkerer alle desimaler i FREQ-variabelen før den brukes. Er verdien på FREQ-variabelen 0 eller mindre, teller den ikke med i statistiske beregninger.

Feilmeldinger i REPORT

I vår omgang med REPORT kan det hende vi får en feilmelding i SASLOG'en istedenfor en tabell. De fleste feilmeldingene er forståelige, imidlertid er det noen feilmeldinger som kan være vanskelige å tolke:

Statistiske beregninger uten analysevariabel

Skal vi gjøre statistiske beregninger, *må* vi gjøre dette på analysevariable. I programmet under har vi koblet sum til kjonn som ikke er noen analysevariabel. Vi får da en feilmelding:

```
PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
  COLUMN naering kjonn vekt1996-vekt1999;
  DEFINE naering / GROUP WIDTH=6 SPACING=0;
  DEFINE kjonn / ACROSS WIDTH=5 ;
  DEFINE vekt1996 / ANALYSIS SUM WIDTH=10 ;
  DEFINE vekt1997 / ANALYSIS SUM WIDTH=10 ;
  DEFINE vekt1998 / ANALYSIS SUM WIDTH=10 ;
  DEFINE vekt1999 / ANALYSIS SUM WIDTH=10 ;
  TITLE 'Rapport 3. Gruppering etter næring og kjønn' ;
RUN;
```

```
ERROR: There is a statistic but no ANALYSIS usage associated
with the column defined by the following elements.
Name                               Usage
-----
sum
```

Ved å koble sum til analysevariablene isteden vil programmet gå bra:

```
COLUMN naering kjonn (vekt1996-vekt1999),sum;
```

Flere statistiske beregninger knyttet sammen samtidig

Vi kan ikke knytte flere typer statistiske beregninger til samme variabel i COLUMN-setningen. Det gir feilmelding:

```
COLUMN naering kjonn n, (vekt1996-vekt1999), sum;
```

```
ERROR: There is more than one statistic associated
with the column defined by the following elements.
Name                               Usage
-----
n                                   STATISTIC
Vekt1996                           ANALYSIS
sum                                  STATISTIC
```

Her kan vi sette N inne i en parentes sammen med SUM:

```
COLUMN naering kjonn (vekt1996-vekt1999), (n sum);
```

Ingen data å lage tabell av

Hvis en observasjonene mangler verdi for en av grupperingsvariablene, utelates som kjent hele observasjonen fra tabellen (såfremt vi ikke har med MISSING-option i PROC REPORT-setningen). Mangler alle observasjonene på datasettet verdi for minst en av grupperingsvariablene, vil altså alle observasjonene holdes utenfor tabellen og det blir ingen observasjoner igjen å lage tabell for. Denne meldingen vil da komme i SASlog:

```
WARNING: A GROUP, ORDER, or ACROSS variable is missing on every
observation.
```

Ved å ta med MISSING-option i PROC REPORT-setningen vil det bli laget en tabell. Denne bør sjekkes nøye for å finne ut hvilken effekt de manglende verdier har på resultatet i tabellen.

```
PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING;
```

Tabeller skrevet direkte til html-fil

HTML (Hyper-Text Markup Language) er et format som brukes for å merke innholdet i en fil med hensyn på layout. Vi kan merke hva som er overskrift, tabellhode, forspalte, tabellceller etc. Med denne merkingen forteller vi hvordan de forskjellige delene av tabellen skal vises. Når filen er lagret i html-format, vil den enkelt kunne legges ut på Internett, og interesserte kan bruke en nettleser til å se på tabellen. Nettleseren tolker html-formatet og viser en pen tabell på skjermen.

Vi kan bruke SAS til å lage html-filer i stedet for vanlig utskrift. Til dette formålet bruker vi SAS sin ODS (Output Delivery System). Dette er et system som behandler utskrift fra SAS og gjør den om til forskjellige fil-formater, f.eks. html, xml og rtf. Før tabellprogrammet angir vi hva slags utskriftstype vi ønsker og navn på filen utskriften skal styres til. Etter tabellprogrammet avslutter vi utskrift til filen med en CLOSE-kommando. Vi får som standard ut html-filer med SSB sine farger.

Her et eksempel som gir en html-tabell med vårt standard utseende:

```
ODS html FILE='bef2.html';
PROC TABULATE DATA=bef FC='-----' NOSEPS MISSING ORDER=DATA;
  WHERE aar='1999' AND landsdel IN('1','2','3','4');
  CLASS kjonn landsdel /PRELOADFMT;
  VAR innbyggere dode innvandrede;
  TABLE (ALL='Antall'*SUM=' '*F=7.0
    ALL='Prosent'*COLPCTSUM=' '*F=numx7.1)*
    (ALL='I alt'
    landsdel=' ')
    ,
    (ALL='I alt'      kjonn=' ')*
    (innbyggere='Innbyggere'
    dode='Døde'
    innvandrede='Innvandrede')
  / RTS=17 MISSTEXT='- ' BOX='Landsdel' ;
FORMAT kjonn      $kjonn.
        landsdel  $ldel.
        aldersgruppe $aldgrp.
        ;
TITLE 'Befolkning etter landsdel og kjønn. ';
FOOTNOTE 'Kilde: SSB' ;
RUN;
ODS html CLOSE;
RUN;
```

Merk ellers fotnoteteksten som bør være med.

Hvis vi bruker Windows SAS som klient og unix SAS som tjener, ønsker vi gjerne å få overført html-filen til Windows NT. Da kan vi bruke PROC DOWNLOAD til dette:

```
PROC DOWNLOAD
  INFILE='bef2.html'
  OUTFILE='h:\krl\befolkning\bef2.html';
RUN;
```

Når vi jobber direkte mot Unix, overfører vi html-filen ved hjelp av et FTP-program.

Utskrift fra SAS til Excel eller Word

Det kan være flere grunner til at vi ønsker å ta en SAS-utskrift inn i et Excel regneark: det er for eksempel mulig å 'videreforedle' tallene i tabellen ved hjelp av Excel.

For å få SAS-utskriften inn i Excel, lønner det seg å gå veien om en html-fil, se forrige kapittel. Når tabellen er lagret som en html-fil, kan vi enkelt åpne den i Excel:

- Sørg for å overføre html-filen til Windows.
- Start Excel
- Velg Fil og Åpne
- I dialogvinduet som nå er åpnet, velg filtype HTML-dokumenter, eventuelt Alle filer. Velg filen med å dobbeltklikke på den

Tilsvarende framgangsmåte kan vi bruke om vi ønsker tabellen satt inn i et Word-dokument. En alternativ måte å få tabellene inn i Excel er beskrevet i neste kapittel.

Behandle unix-filer fra windows

Vi har vi metode for å se på unix-filer fra nettleseren på windows (Internet Explorer). Ved å bruke denne metoden slipper vi å overføre filen til windows. Før denne metoden kan brukes må vi gjøre en liten engangsjobb. Den går ut på at vi først må lage et område under hjemmeområdet vårt som heter `www_docs`. Dette området skal vi senere legge tabellene våre på. For å lage området logger vi først på en unix-maskin. Deretter skriver vi disse kommandoene:

```
cd
mkdir www_docs
cd ..
chmod o+x ini
```

`ini` er initialene dine. Kommandoene gjør følgende: `cd` går til ditt hjemmeområde, `mkdir` lager området `www_docs`, `cd ..` går et nivå opp i filtreet, `chmod` gir alle eksekveringstillatelse for hjemmeområdet ditt. Den siste kommandoen må gjøres for å få tillatelse til å se på unix-filene som legges under `www_docs` fra Internet Explorer. Denne åpningen utgjør en liten sikkerhetsrisiko da også andre kan se på dine tabeller og utføre eksekveringskommandoer på hjemmeområdet ditt. For å gjøre det vanskeligere å komme inn på et område kan vi legge en sperre på det. Det gjør vi på unix ved å lage en fil som skal hete `.htaccess`. Filen legges under området `www_docs` og kan se slik ut:

```
AuthName "Sperring av hjemmekatalog"
AuthType Basic
AuthUserFile /www/conf/passwd
require user ini
IndexOptions FancyIndexing
IndexOptions ScanHTMLTitles
```

`ini` er fortsatt initialene dine. Hvis andre skal ha tilgang til området ditt, kan du legge til flere brukere etter din egen, skilt med en blank. Kommandoene for å lage filen er disse:

```
cd
cd www_docs
```

nedit .htaccess

Den siste kommandoen starter en editor som heter *nedit*. Her skriver du inn teksten til *.htaccess*. Når det er gjort, lagrer du og går ut av *nedit*.

På unix er det også laget en standard html-layout som passer med tabellene på SSBs web-sider. Den heter *publ01* og har også en engelsk variant; *publ01e*.

Eksempel som viser hvordan vi lager en tabell med publiseringsmalen og hvordan vi lagrer den på *www_docs*-området vårt på unix.

```
ODS html file='$HOME/www_docs/tabell29.html' style=publ01;
PROC TABULATE DATA=bef FC='-----' F=8. NOSEPS MISSING;
  CLASS landsdel ;
  VAR innbyggere ;
  TABLE landsdel='
      '
      'innbyggere*SUM='
      /BOX='Landsdel';
  FORMAT landsdel $ldel.;
  TITLE 'Tabell 29. Antall innbyggere etter landsdel' ;
RUN;

ODS html CLOSE;
RUN;
```

\$HOME er hjemmeområdet på unix til den som er logget på. For å se på denne tabellen fra Internet Explorer skriver vi dette i adressefeltet i Internet Explorer:

<http://www.ssb.no/~ini/tabell29.html>

Nok en gang er *ini* initialene dine. Første gang du skal se på en tabell i Internet Explorer etter at den er startet, får du spørsmål om unix-bruker og unix-passord (om du har laget *.htaccess*-filen). Selv om adressen inneholder www.ssb.no ligger ikke filen tilgjengelig på internett. Vi legger også merke til at vi ikke skal ha med *www_docs* i referansen.



Denne filen kan også åpnes i excel. Kopier adressen fra Internet Explorer, velg *Fil, Åpne* i Excel og lim inn adressen i feltet for filnavn. Her vil du også bli spurt om unix-bruker og unix-passord første gang. Tabellen blir ikke så veldig pen i Excel. Det er laget en konverteringsfunksjon for å få penere Excel-regneark, denne heter *saxls* og kalles ved å legge *saxls* i filreferansen når den skal åpnes i Excel:

<http://www.ssb.no/saxls/~ini/tabell29.html>. Tabellen vises uten rutemønster, det legger vi på ved å velge *Verktøy, Alternativer* i Excel og krysse av for rutenett.

For å få venstrestilt forspalte fra REPORT når vi bruker publiseringsmalen, trenger vi å legge til en *STYLE*-opsjon i *DEFINE*-setningen i REPORT:

```
ODS html file='$HOME/www_docs/rapport16.html' style=publ01;
PROC REPORT DATA=aku HEADLINE HEADSKIP NOWINDOWS MISSING ;
  COLUMN naering vekt1998-vekt1999;
  DEFINE naering / GROUP WIDTH=23 SPACING=0 STYLE=[htmlclass="level1"];
  DEFINE vekt1998 / ANALYSIS SUM WIDTH=8 '1998' F=8.;
  DEFINE vekt1999 / ANALYSIS SUM WIDTH=8 '1999' F=8.;
  RBREAK BEFORE / SUMMARIZE SKIP;
  FORMAT naering $nar.;
  TITLE 'Rapport 16. Sysselsatte gruppert etter naering. ' ;
RUN;
ODS html CLOSE;
RUN;
```

Definisjoner av statistiske nøkkelord

Følgende notasjon brukes i formlene:

x_i	Observasjon nummer i av en variabel blant de som ikke er missing
w_i	Vekten til observasjon x_i , hvis WEIGHT-setningen er brukt, ellers 1.
n	Antall observasjoner som ikke har missing-verdier for en variabel
\bar{x}	$= \sum w_i x_i / \sum w_i$
d	$= n$ når VARDEF=N VARDEF, se side 39
	$= n - 1$ når VARDEF=DF (standard)
	$= \sum w_i$ når VARDEF=WEIGHT
	$= \sum w_i - 1$ når VARDEF=WDF
s^2	$= \sum w_i (x_i - \bar{x})^2 / d$
s	$= \sqrt{s^2}$

Disse statistiske beregninger kan utføres i TABULATE:

CSS	$\sum (w_i (x_i - \bar{x}))^2$, veid kvadratavvik
CV	$100s / \bar{x}$, variasjonskoeffisient, prosent
KURTOSIS	$\sum \sqrt{z_i^4 \times n(n+1)/(n-1)(n-2)(n-3) - 3(n-1)^2/(n-2)(n-3)}$, måler halens tyngde, dvs. om noen verdier er veldig langt fra gjennomsnittet.
MAX	maksimumsverdi
MEAN	\bar{x} , aritmetisk gjennomsnitt
MIN	minimumsverdi
N	antall observasjoner som har verdi
NMISS	antall observasjoner som mangler verdi
PCTN	prosent beregnet av antall observasjoner, se Prosenttabeller, side 10
PCTSUM	prosent beregnet av forholdet mellom en sum og totalsummen, se Prosenttabeller, side 10.
PROBT	tosidig p -verdi for Student's t , med $n-1$ frihetsgrader; sannsynligheten under nullhypotesen av å få en absoluttverdi for t større enn t -verdien observert fra utvalget.
RANGE	MAX - MIN, avstanden mellom største og minste verdi
SKEWNESS	$\sum z_i^3 \times n(n-1)(n-2)$, måler skjevheten.
STD	s , standardavviket
STDERR	$s / \sqrt{\sum w_i}$, standardfeil (for gjennomsnittet).
SUM	$\sum w_i x_i$, veid sum
SUMWGT	$\sum w_i$, summen av vektene
T	$t = \bar{x} \sqrt{(n)/s}$, student's t for H_0 : populasjonsgjennomsnitt = 0
USS	$\sum w_i x_i^2$, veid kvadratsum
VAR	s^2 , variansen

Vedlegg

Innlesingsprogram for befolkningsdatafilen bef

```

DATA bef;
INFILE CARDS TRUNCOVER;
INPUT @01 Aar          $CHAR4.
      @06 Landsdel    $CHAR1.
      @08 Kjonn       $CHAR1.
      @10 Aldersgruppe $CHAR1.
      @12 Innbyggere  6.
      @19 Dode        5.
      @25 Innvandrede 4.
;
LABEL aar             = 'År'
      landsdel        = 'Landsdel'
      kjonn           = 'Kjønn'
      aldersgruppe    = 'Aldersgruppe'
      innbyggere      = 'Innbyggere pr. 1.1'
      dode             = 'Døde'
      innvandrede     = 'Innvandrede'
;
CARDS;
1998 1 1 1 268801 123 2675
1998 1 1 2 676477 2678 7304
1998 1 1 3 118893 8349 109
1998 1 2 1 255427 89 3216
1998 1 2 2 673213 1536 7324
1998 1 2 3 177748 9969 112
1998 2 1 1 91465 43 807
1998 2 1 2 186819 680 2072
1998 2 1 3 29190 2034 43
1998 2 2 1 86731 28 839
1998 2 2 2 180448 357 1609
1998 2 2 3 43393 2243 42
1998 3 1 1 143060 76 909
1998 3 1 2 317029 1090 2342
1998 3 1 3 56499 3899 32
1998 3 2 1 135932 36 981
1998 3 2 2 302484 567 2206
1998 3 2 3 82758 4382 35
1998 4 1 1 80873 44 642
1998 4 1 2 184385 751 1359
1998 4 1 3 31615 2300 18
1998 4 2 1 76443 40 616
1998 4 2 2 172922 364 1392
1998 4 2 3 44994 2434 20
1998 9 9
1999 1 1 1 271854 141 3272
1999 1 1 2 682564 2610 7497
1999 1 1 3 118959 8560 112
1999 1 2 1 258518 106 3618
1999 1 2 2 678898 1642 7682
1999 1 2 3 177464 10297 147
1999 2 1 1 92136 68 1088
1999 2 1 2 189622 680 2079
1999 2 1 3 29231 2084 32
1999 2 2 1 87518 36 1122
1999 2 2 2 182583 355 1937
1999 2 2 3 43454 2329 44
1999 3 1 1 143880 69 1276
1999 3 1 2 318765 1161 2607
1999 3 1 3 56572 3993 36
1999 3 2 1 136691 43 1193
1999 3 2 2 304460 557 2440
1999 3 2 3 82555 4489 41
1999 4 1 1 80763 40 1006
1999 4 1 2 183487 695 1841
1999 4 1 3 31726 2325 18
1999 4 2 1 76268 26 969
1999 4 2 2 172375 344 1761
1999 4 2 3 44986 2520 23
1999 9 9
;
RUN;

```

Innlesingsprogram for arbeidskraftundersøkelsens (AKU) datafil aku

```

DATA aku;
  INFILE CARDS TRUNCOVER;
  INPUT @01 Naering          $CHAR2
        @04 Landsdel        $CHAR1
        @06 Kjonn           $CHAR1
        @08 IArbeid1996     $CHAR1
        @10 IArbeid1997     $CHAR1
        @12 IArbeid1998     $CHAR1
        @14 IArbeid1999     $CHAR1
        @16 Vekt1996         6.
        @23 Vekt1997         6.
        @30 Vekt1998         6.
        @37 Vekt1999         6.
        @44 Arbtid1996       5.2
        @50 Arbtid1997       5.2
        @56 Arbtid1998       5.2
        @62 Arbtid1999       5.2
  ;
  LABEL Naering             = 'Næring'
        Landsdel           = 'Landsdel'
        Kjonn              = 'Kjønn'
        IArbeid1996        = 'I arbeid i 1996'
        IArbeid1997        = 'I arbeid i 1997'
        IArbeid1998        = 'I arbeid i 1998'
        IArbeid1999        = 'I arbeid i 1999'
        Vekt1996            = 'Vekt 1996'
        Vekt1997            = 'Vekt 1997'
        Vekt1998            = 'Vekt 1998'
        Vekt1999            = 'Vekt 1999'
        Arbtid1996          = 'Arbeidstid 1996'
        Arbtid1997          = 'Arbeidstid 1997'
        Arbtid1998          = 'Arbeidstid 1998'
        Arbtid1999          = 'Arbeidstid 1999'
  ;
  CARDS;
01 1 1 1 1 1 1 045436 046664 047363 048323 37.50 37.50 37.50 57.50
02 1 2 2 2 1 1 033331 034346 034241 038811 . . 37.50 37.50
05 3 2 1 1 1 1 055525 056723 056352 058100 23.50 24.00 23.50 37.50
05 2 1 2 1 2 2 044240 045424 042240 045635 . 42.00
10 3 1 1 2 1 1 045436 046664 047363 046323 63.00 . 57.00 59.50
10 4 1 1 1 1 1 045436 046664 047363 049323 37.50 39.50 41.50 38.00
11 1 1 2 2 1 1 033331 034346 034241 038811 . 37.50 37.50
12 2 1 1 1 1 1 045436 046664 047363 041323 37.50 37.50 37.50 37.50
13 1 1 2 1 1 1 033331 034346 034241 038811 38.50 . 37.50 37.50
14 3 2 1 1 2 1 055525 056723 056352 053100 37.50 37.50 . 37.50
15 1 2 2 1 2 2 044240 045424 042240 045635 . 37.50
16 1 1 2 1 1 1 045436 046664 047363 048323 37.50 . 37.50 37.50
16 3 2 1 2 1 1 075436 077664 075363 074423 34.50 . 37.50 37.50
17 2 1 2 1 1 1 033331 034346 034241 048811 . 37.50 37.50 37.50
18 1 2 1 1 1 1 045436 046664 047363 048323 37.50 37.50 37.50 37.50
19 1 1 2 2 1 1 033331 034346 034241 038811 . 39.50 43.50
20 2 2 1 1 1 1 055525 056723 056352 053100 37.50 37.50 37.50 37.50
21 1 2 2 2 2 2 044240 045424 042240 045635
22 1 1 1 1 1 1 045436 046664 047363 049323 37.50 37.50 37.50 37.50
23 1 1 2 2 1 1 033331 034346 034241 038811 . 47.50 37.50
24 1 2 1 1 1 1 055525 056723 056352 053100 37.50 37.50 38.50 37.50
25 1 2 1 1 1 1 045436 046664 047363 049323 37.50 37.50 38.50 37.50
25 3 1 1 1 1 1 055436 056664 057363 057723 47.50 37.50 38.50 37.50
25 4 1 1 1 1 1 055436 056664 057363 056323 37.50 37.50 37.50 37.50
26 2 1 2 2 2 1 033331 034346 034241 048811 . 41.50
27 4 2 1 2 2 2 057777 057635 055673 057000 37.50
27 3 2 1 1 1 1 055525 056723 056352 053100 33.50 37.50 39.50 47.50
28 1 1 1 1 1 1 045436 046664 047363 . 37.50 37.50 37.50 37.50
29 2 1 2 2 1 1 033331 034346 034241 038811 . 39.50 47.50
30 1 2 1 1 2 1 055525 056723 056352 057100 34.50 37.50 39.50 37.50
31 1 2 2 2 2 2 044240 045424 042240 045635
32 2 1 1 2 1 1 033331 034346 034241 048811 . 37.50 37.50 47.50
32 1 1 1 1 1 1 066625 065466 068661 067133 32.50 . 37.50 37.50
33 2 2 1 1 2 1 055525 056723 056352 057100 37.50 42.50 . 37.50
33 4 2 1 1 1 1 055525 056723 056352 058100 33.50 37.50 42.50 44.50
34 1 1 2 2 2 2 044240 045424 042240 049635
35 3 1 1 1 1 1 045436 046664 047363 048323 37.50 38.00 41.50 47.50
36 1 1 1 2 1 1 033331 034346 034241 049811 27.50 . 42.50 37.50
37 3 1 1 1 1 1 045436 046664 047363 048323 37.50 37.50 37.50 35.50
40 4 1 2 2 1 1 033331 034346 034241 038811 . 37.50 37.50
41 3 2 1 1 2 1 055525 056723 056352 056100 37.50 37.50 .
45 4 1 1 1 1 1 044240 045424 042240 048635 32.50 32.50 32.50 32.50
30 1 . 044233 045635 043321 048982 . 37.50 37.50 37.50
45 3 . 034233 033635 035321 035982 37.50 37.50 37.50
  ;
  RUN;

```

Programmet som lager tekstformater som brukes

```
PROC FORMAT ;
  VALUE $kjonn (NOTSORTED)
    '2' = 'Kvinner'
    '1' = 'Menn'
    OTHER = 'Uoppgitt'
  ;

  VALUE $ldel (NOTSORTED)
    '4' = 'Nord'
    '3' = 'Vest'
    '1' = 'Øst'
    '2' = 'Sør'
    OTHER = 'Uoppgitt'
  ;

  VALUE $aldgrp
    '1' = '0-19 år'
    '2' = '20-67 år'
    '3' = '67 år og over'
    OTHER = 'Uoppgitt'
  ;

  VALUE $svar (NOTSORTED)
    '1' = 'Ja'
    '2' = 'Nei'
    OTHER = 'Uoppgitt'
  ;
RUN;
```

Stikkordregister

A

ALL..... 5,6,10,41
 analysevariabel..... 4,5,37,38,39,42,64,68
 andelstall 27
 antall observasjoner..... 10,13
 ASCENDING..... 40

B

BOX 7,43
 BREAK 62,66
 BY 38,39,43,62,64
 BYVAL..... 18
 BY-variabel..... 64

C

CALL DEFINE 62,68
 CLASS 3,4,20,35,36,38,40,41,49
 CLASSDATA 38
 CLASSLEV..... 38,40
 COLPCTN 42,48
 COLPCTSUM..... 11,42
 COLUMN 62,64
 COMPUTE 62,67
 CONDENSE 43
 CONTENTS..... 38
 CSS 41,64,74
 CV 41,64,74

D

DEFINE 65
 DEFINE 62
 denominator 10
 DESCENDING 40

E

ENDCOMP 62,69
 Excel 72
 EXCLNPWGHTS 38,46
 EXCLUSIVE 38,40

F

FC..... 4,38
 feilmeldinger 47,69
 FORMAT..... 6,38,41,62,69
 FORMCHAR 38,62
 forspalte..... 3,4
 forspaltehode..... 3,43
 FREQ 22,38,41,62,69
 FUZZ..... 43

G

GROUPINTERNAL 40
 grupperingsvariabel
 4,5,35,38,39,40,41,42,43,49,63,64,65,67

H

hode..... 3
 html 71

I

Ikke-hierarkiske tabeller..... 29
 INDENT..... 43

K

KEYLABEL..... 38,41
 KEYWORD 38,41
 KURTOSIS 41,74

L

LABEL..... 38,41,43
 LCLM 41
 LINE 68
 LINE 62

M

manualer..... 3
 MAX 4,41,64,74
 MEAN..... 4,41,64,65,74
 medianverdi..... 42
 MIN..... 4,41,64,74
 MISSING
 4,5,21,24,25,35,37,39,40,43,46,49,63,64,66
 MISSTEXT 7,43
 MLF 40

N

N 4,5,21,22,37,39,41,46,64,69,74
 nevner..... 10,42,48
 NMISS 22,37,41,46,64,74
 NOCONTINUED..... 43
 NOSEPS..... 4,39
 NOTRAP..... 39

O

ODS 71
 oppblåsing 22,46,69
 ORDER 39,40
 OUT 39

P

P142
 P10 42
 P542
 P90 42
 P95 42
 P99 42
 PAGEPCTN 42
 PAGEPCTSUM 42
 PCTN 4,10,13,22,41,46,48,64,74
 PCTSUM 4,10,13,41,46,64,74

PRELOADFMT	40	STD	4,41,64,74
PRINTMISS	43	STDERR	41,64,74
PROBT	41,64,74	STYLE	39,40,46
PROC DOWNLOAD	71	SUM	4,5,21,41,65,74
PROC FORMAT	6	SUM:	64
PROC FREQ	34,46	SUMWGT	22,41,64,74
PROC MEANS	32	T	
prosentiler	41	T	41,64,74
prosentuering	10,48,64	tabellinnmat	3
Q		tabelloverskrift	3
Q1	42	TABLE	4,38,41,42,43,49
Q3	41	TITLE	4,38,46
QRANGE	42	tusenskille	17
R		U	
RANGE	41,64,74	UCLM	41
ratetabeller	28	USS	41,64,74
RBREAK	67	V	
REPPCTN	42	VAR	3,4,20,38,46,49,64
REPPCTSUM	42	VAR (varians)	41,74
ROW	45	VARDEF	39,64,74
ROWPCTN	42	varians	64
ROWPCTSUM	12,42	veing	3,22,41,46,69
RTS	7,46	W	
S		WEIGHT	3,22,38,41,46,62,69
SKEWNESS	41,74	WHERE	10,38,39,62,64
skilletegn	4	Word	72
statistiske nøkkelord	4,41,64,74		

B Returadresse:
Statistisk sentralbyrå
NO-2225 Kongsvinger

Statistisk sentralbyrå

Oslo:

Postboks 8131 Dep
NO-0033 Oslo
Telefon: 21 09 00 00
Telefaks: 21 09 49 73

Kongsvinger:

NO-2225 Kongsvinger
Telefon: 62 88 50 00
Telefaks: 62 88 50 30

E-post: ssb@ssb.no
Nett: www.ssb.no