

Pia E. Tønjum

Notater

**Dokumentasjon av edb-rutiner
for kjeding av årlig og
kvartalsvis nasjonalregnskap**

Innholdsfortegnelse

1	EDB-rutiner for kjeding av års- og kvartalstall	3
	1.1 Innledning	3
	1.2 Generelt	4
	1.3 Litt om ny og gammel metode	4
2	Hovedprogrammene	6
	2.1 Styreprogrammet \$kjed.....	10
	2.2 Blokkindeligen.....	12
	2.3 Generelle kjedeformler	13
	2.4 Andre subrutiner som benyttes av hovedprogrammene	14
	2.5 Spesielle problemer med hensyn til metode og beregning av unntak.....	15
	2.6 Spesialrutiner for beregning av fastpristall i kvartalsregnskapet	15
	2.7 Kontroll av resultatene	17
	2.8 Når databanken er ferdig kjedet.....	17
3	Valg av nytt referanseår.....	19
	3.1 Styreprogrammet \$velg_refaar	22
	3.2 Generelle omregningsformler for nytt referanseår	23
	3.3 Uttak av data med nytt referanseår	24
4	Generelt om uttak av data	24
5	Referanser	27
6	Vedlegg: Utskrift av programkode	
	Vedlegg 1: Styreprogrammet \$kjed.....	28
	Vedlegg 2: Kjedeformelen \$kjed_sektor.....	46
	Vedlegg 3: Styreprogrammet \$velg_refaar.....	52
	Vedlegg 4: Omregningsformel for valgfritt referanseår \$kjed_valgt_refaar.....	64
	De sist utgitte publikasjonene i serien Notater	66

1 EDB-rutiner for kjeding av års- og kvartalstall

1.1 Innledning

Dette er en teknisk dokumentasjon av edb-rutinene for kjeding av årlige og kvartalsvise nasjonalregnskapstall. Hovedvekten er lagt på å gjøre rede for hvordan prosedyrene er programmert, hva de gjør, hvor de henter data, og hvordan de vedlikeholdes. En mer beregningsteknisk beskrivelse av metoder for kjeding er gitt i NOS Nasjonalregnskap 1991 (kapittel 4) og i NOS Kvartalsvis nasjonalregnskap 1978-1990 (avsnitt 4.2.3).

Etter en generell oversikt over når og hvorfor det er aktuelt å benytte kjedingsrutinene, beskrives forskjellene mellom gammelt og nytt system med hensyn til metode (leseren kan evt. hoppe over avsnitt 1.3). Deretter beskrives bruk og vedlikehold av hovedprogrammet og subrutinene. I den forbindelse vil også spesialrutinene for beregning av fastpristall i kvartalsregnskapet bli beskrevet. Hensikten er å gjøre rede for de tilpasninger som er foretatt for å kjede tilbakegående tidsserier for eksport og import i forbindelse med Marsregnskapet. Avslutningsvis i kapittel 2 omtales uttak av kjedede data. Etter ønske fra bl.a. OECD, IMF, EUROSTAT og mange andre brukere av nasjonalregnskapstall, er det også laget prosedyrer som gjør det mulig for brukeren å velge nytt referanseår selv. Disse prosedyrene blir beskrevet del 3. Til slutt gjennomgås uttak av nasjonalregnskapstall generelt, fordi arbeidsdatabasene som er omtalt i dette notatet, ofte må åpnes i en bestemt rekkefølge for at brukeren skal få ut «riktige» vekstrater (i databasene med offisielle tall får brukeren automatisk tak i de «riktige», avstemte volumvekstratene, slik at åpningsrekkefølgen er av mindre betydning). Med mindre annet er nevnt eksplisitt, er beregning av års- og kvartalstall (kjedingsrutiner for kvartalstall er testet, men ikke tatt i bruk) programmert på samme måte.

Alle prosedyrer og hjelpevariable har forøvrig fått mest mulig selvforklarende navn. Dessuten inneholder alle prosedyrer forklaringer og dokumentasjon av beregningene som utføres mv. Styreprogrammene \$kjed og \$velg_refaar er dessuten bygget opp på samme måte, og alle prosedyrer er navngitt likt, f.eks. utførerer \$kjed_kons og \$velg_refaar_kons beregninger for de samme konsumvariablene (se nærmere omtale i kapittel 2 og 3). Enkelte hjelpeprosedyrer benyttes også av begge programmene.

Samtlige prosedyrer er forøvrig programmert i FAME (Forecasting, Analysis, and Modeling Environment). Alle databaser som prosedyrene leser fra eller oppretter når de kjøres, er tids-seriedatabaser i FAME, og alle variable er navngitt i henhold til nasjonalregnskapets standard navnestruktur, jfr. Teknisk dokumentasjon av realregnskapets FAME-databaser (EiP 1996, STo og PET 1998), samt års- og kvartalsregnskapenes kontoplan. For en generell innføring i UNIX og FAME vises til Sjøberg (1995) og Vogt (1994).

Rapporten er først og fremst beregnet på driftsansvarlige som skal foreta kjøring og vedlikehold av års- og kvartalsregnskapenes rutiner, og forutsetter at man har god kjennskap til NR og KNR's produksjonssystem, samt UNIX og FAME. Forhåpentligvis vil også deler av rap-

porten ha mer almen interesse, deriblant kapitlene om metodevalg og bruk av databasene for uttak av data.

1.2 Generelt

Kjeding er nødvendig for å få tidsserier med sammenlignbare fastpristall, dvs. i samme prissett. I kvartalsregnskapet skiftes basisåret (normalt) hver vår når nye årsregnskapstall tas inn som modellgrunnlag. De nye årsregnskapstallene overføres fra Oracle-databasen, som inneholder fastpristall i t-1 priser, til FAME-databasene nr01, nr02, knrgrlag og knrgrlag_agg. Sistnevnte database inneholder formler, nr02 inneholder formler og tidsserier, mens de to øvrige bare inneholder tidsserier (hvilke variable som er lagret i de forskjellige databasene, samt overførings- og aggregeringsrutinene er dokumentert av EiP, se kapittel 3 i Teknisk dokumentasjon av realregnskapets FAME-databaser (EiP 1996, STo og PET 1998) Alle databasene ligger på område \$NASJREGN/tidsserier/NR/DB/.

Etter at KNR-modellen er oppdatert og Marsregnskapet er ferdig beregnet, avstemmes kvartalstallene mot årlig nasjonalregnskap i løpende og faste priser. Til avstemmingen benyttes kjedede årstall. Årstallene har samme basisår som KNR-modellen benytter.

Metoder for avstemming er forøvrig beskrevet i avsnitt 4.2.4 i NOS Kvartalsvis nasjonalregnskap 1978 - 1990, Interne Notater 85/15 og Interne Notater 85/22. Se også Rutiner for avstemming av kvartalstall mot årstall (JOu 1995), Litt om avstemming av kvartalstall mot årstall (JOu 1996), samt Teknisk dokumentasjon av beregningsopplegg for KNR (EiP 1996, STo og PET 1998).

1.3 Litt om ny og gammel metode

I dette avsnittet beskrives bakgrunnen for valg av metode, og ny og gammel metode blir sammenlignet. Leseren kan gå videre til neste avsnitt dersom metodevalg mv. er av mindre interesse.

Tidligere ble kvartalstallene omregnet til årsregnskapets basisår før de ble avstemt, og kjedet til nytt referanseår etter avstemmingen. I utgangspunktet ble kjedingsrutinene programmert med tanke på dette. Etter Hovedrevisjonen har man ved seksjon 210 isteden valgt å avstemme kvartalstallene mot kjedede årsregnskapstall, fordi vi i første omgang ikke hadde kvartalsdata lenger tilbake enn til 1993 (etter ny revidering av kontoplanen) og modellen skulle kjøres med 1993 som basisår tilbake til 1978. Beslutningen ble tatt etter at programmeringsarbeidet delvis var fullført. En del tilpasninger ble derfor foretatt, slik at programmene nå kan brukes til å kjede om års- og eventuelt kvartalstall i t-1 priser til basisårets priser. EUROSTAT har tidligere bedt om å få kvartalstall med samme basisår som årsregnskapet. Foreløpig er det ikke opprettet en kvartalsdatabase med t-1 priser, men det vil trolig bli aktuelt, fordi alle nasjonalregnskapstall, både på detaljert og aggregert nivå, skal kjedes direkte (med unntak av de få variable som er omtalt i neste avsnitt). Kjdede tall skal med andre ord ikke summeres uansett

nivå. Fordi alle aggregater kjedes direkte, vil man få kjedingsavvik i faste priser dersom man forsøker å summere data eller beregne definisjonssammenhenger ut i fra tabellene (dvs. såkalt ikke-additivitet). Tilgang i alt vil f.eks. ikke bli lik anvendelse i alt. Kjedingsavvik er forøvrig nærmere omtalt i kapitlet om avstemming og kontroll av KNR-tallene i Teknisk dokumentasjon av beregningsopplegg for KNR. I det samme dokumentasjonsnotatet er det også vist et eksempel på kjedingsavvik i vedlegget.

I kvartalsregnskapet ble det tidligere gjort unntak fra den generelle kjedeformelen for enkelte investerings- og eksportvariable. Ved å unntaksbehandle serier som kan være problematiske, ser ofte tallene «rimeligere» ut på detaljert nivå, men dette medfører uoverensstemmelser i tabellene. I kvartalsregnskapet ble problemet tidligere løst ved å benytte to separate databaser. Man hadde en database med kjedede tidsserier og rebaserte summer (knr91), som også Forskningsavdelingen benyttet, og en database med kvartalsserier med årsregnskapets basisår (knr). Sistnevnte database ble benyttet til å ta ut tabeller, og kjedingen ble utført av tabellprogrammet.

Det ble i utgangspunktet bestemt ved seksjon 210 at ingen variable skal unntas fra den generelle kjedeformelen, noe innebar at metoden som ble benyttet til å beregne nasjonalregnskapsstall i faste priser, i første omgang ble endret etter Hovedrevisjonen. Etter ønske fra seksjon 530, ble unntakene korrigert som tidligere, og det ble opprettet to databaser. Seinere ble det imidlertid bestemt at seksjon 210 skulle benytte samme metode, slik at vi nå har felles database. Programmene som benyttes for å beregne unntakene er nærmere omtalt i avsnitt 2.5.

Etter at nyinvesteringer, bruttoinvesteringer, salg av brukt realkapital og kapitalslit var ferdig kjedet, og unntakene for bruttoinvesteringer var korrigert, ble tidligere bygninger og anlegg i alt (art 08b) og maskiner og transportmidler i alt (art 08m) summert for hver KNR-næring. Deretter ble de ferdig kjedede investeringsartene summert for hver KNR-næring. I det nåværende edb-systemet blir korrigerte kvartalstall først summert opp pånytt for å få riktig kvartalsfordeling, og så avstemt mot kjedede årstall. Oppsummeringen foretas bl.a. ved hjelp av aggregerings-formlene.

Foruten enkelte eksportprodukter, bruttoinvesteringsarter og - næringer, ble det også gjort andre unntak fra den generelle kjedeformelen. Tidligere ble bruttoproduktet beregnet som differansen mellom produksjon og produktinnsats i kvartalsregnskapet. Skiftvirkningene (2958) ble da beregnet ved balansering av tilgangssiden mot anvendessiden. Prosedyren **\$skiftv** kan evt. kjøres dersom man ønsker å benytte **gammel metode**. Nå kjedes variablene hver for seg i årsregnskapet, og kvartalsregnskapet avstemmes seinere mot disse seriene. Man får derfor som tidligere nevnt kjedingsavvik i faste priser.

I kvartalsregnskapet ble lagertallene tidligere kjedet på NR-varenivå etter generell formel. Disse årsregnskapstallene ble aggregert til KNR-nivå etter kjeding, og deretter kvartalsfordelt med lagertall fra ukorrxx-databasen (resultater fra opprinnelige KNR-kjøringer) som nøkkel. Til slutt ble lagertallene kvotekorrigert (multiplikativ og additiv kvotekorrigerings). Dersom det manglet NR-varer for enkelte år, ble de resterende produktene deflatert med prisindeksen for samme produkt for leveranse til hjemmemarkedet. I det nåværende edb-systemet kjedes lager etter generell formel på KNR-produktnivå. Man benytter imidlertid samme produkt for (beregnet) leveranse til hjemmemarkedet som kjedeindeks.

2 Hovedprogrammene

Foreløpig kan bare serier på detaljert KNR-sektor nivå, KNR-publiseringsnivå og NR-publiseringsnivå kjedes, da prosedyrekall for kjeding av detaljerte NR-tall enda ikke er lagt inn for alle variable. Hovedprogrammet kan imidlertid enkelt utvides til å omfatte flere serier på ulike aggregeringsnivå. (hvilke prosedyrekall som allerede er lagt inn framgår av programkoden for \$skjed, se Vedlegg 1). Forutsatt at variablene som skal kjedes følger nasjonalregnskapets standard navnestruktur, kan man opprette en ny «namelist» (variabel- eller kodeliste), og legge inn prosedyrekall med tilhørende parametre i styringsprogrammet \$skjed. Kallet på kjedeformel-prosedyren legges inn i den blokken der beregningene skal utføres (se nærmere omtale i avsnitt 2.2). Dersom man legger inn nye lister og prosedyrekall i \$skjed eller foretar endringer andre steder, må imidlertid programmene kompiles. Dette kan eventuelt gjøres ved å kjøre input-fila \$NASJREGN/kjed/kompiler.inp, som **kompilerer samtlige prosedyrer** som benyttes til kjedingen (og til å lage omregningsformler).

Ettersom beregningene foregår ved bruk av en generell formel (se avsnitt 2.3), har det også liten betydning for vedlikehold av programmene om variabelnavn eller koder endres, eller om antall variable utvides. Alle variable i faste priser får etternavnet «vl». Forøvrig er vanlig navnestandard fulgt.

Filene med lister som brukes til å kjede KNR-aggregatene har navn av type tab_agg*.txt, og hentes fra område \$NASJREGN/knr/tabell/. Tilsvarende liste-filer for NR-aggregatene hentes fra område \$NASJREGN/tidsserier/tabell/. Styreprogrammet lager case-serier av disse tekstfilene. Nye tekstfiler må eventuelt opprettes dersom man ønsker tabeller med kjedingsresultater på andre aggregeringsnivåer. Alle lister på detaljert KNR-nivå, som brukes til kjedingen, ligger imidlertid i kontoplan.db på område \$NASJREGN/knr/struktur/. Dersom kontoplanen endres, må alle listene rettes før programmene kan kjøres.

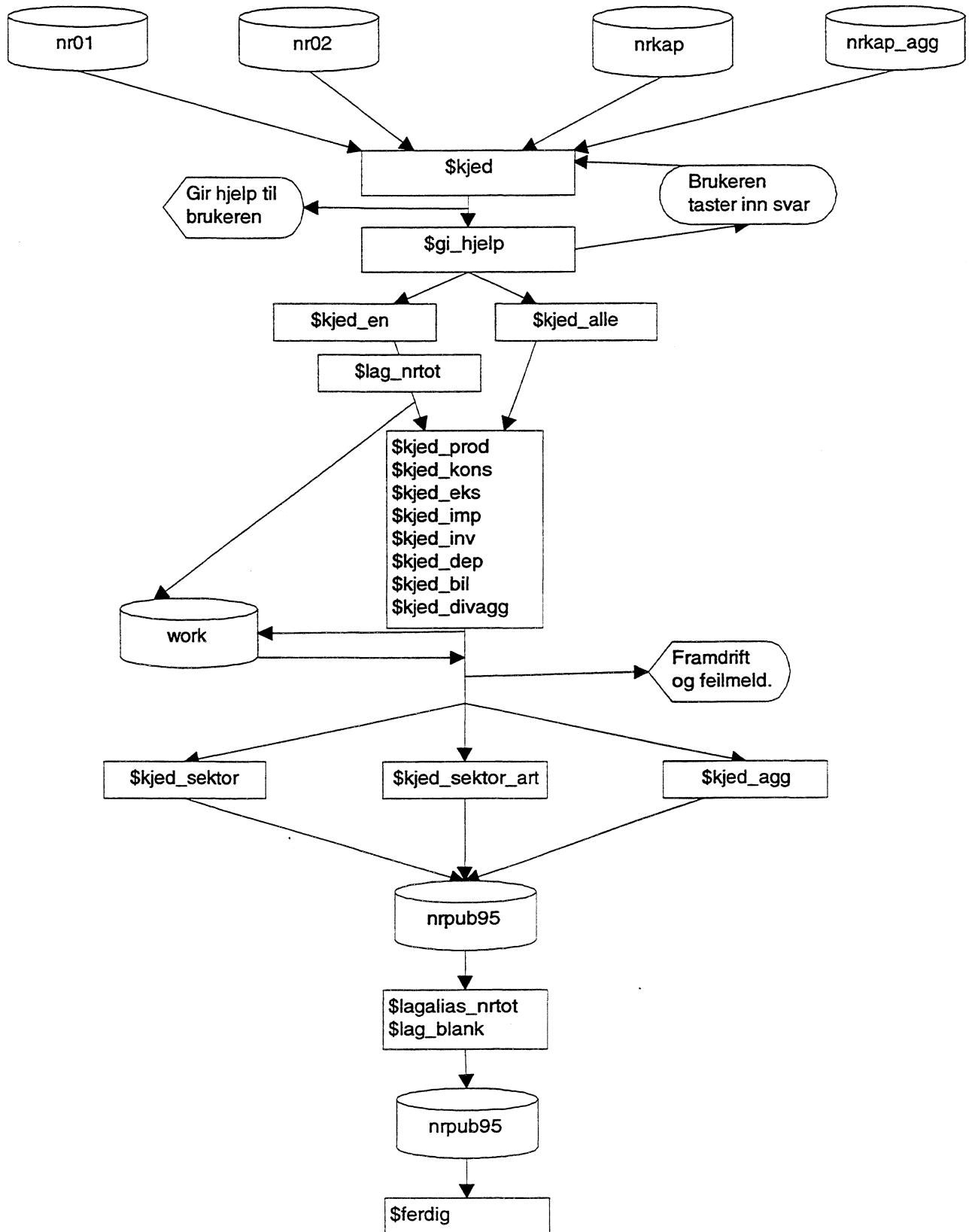
For å unngå avbrudd når rutinene for avstemming av kvartalstall mot årlig nasjonalregnskap kjøres seinere, må databasene med årstall inneholde de samme seriene på detaljert og aggregert nivå som kvartalsdatabasene. Det er derfor lagt inn en del tester i programmene for å sikre at datasettet blir komplett i faste og løpende priser (se forøvrig avsnitt 2.3 om generelle kjedeformler).

Tabell 2.1 Databaser som benyttes av styreprogrammet Skjed

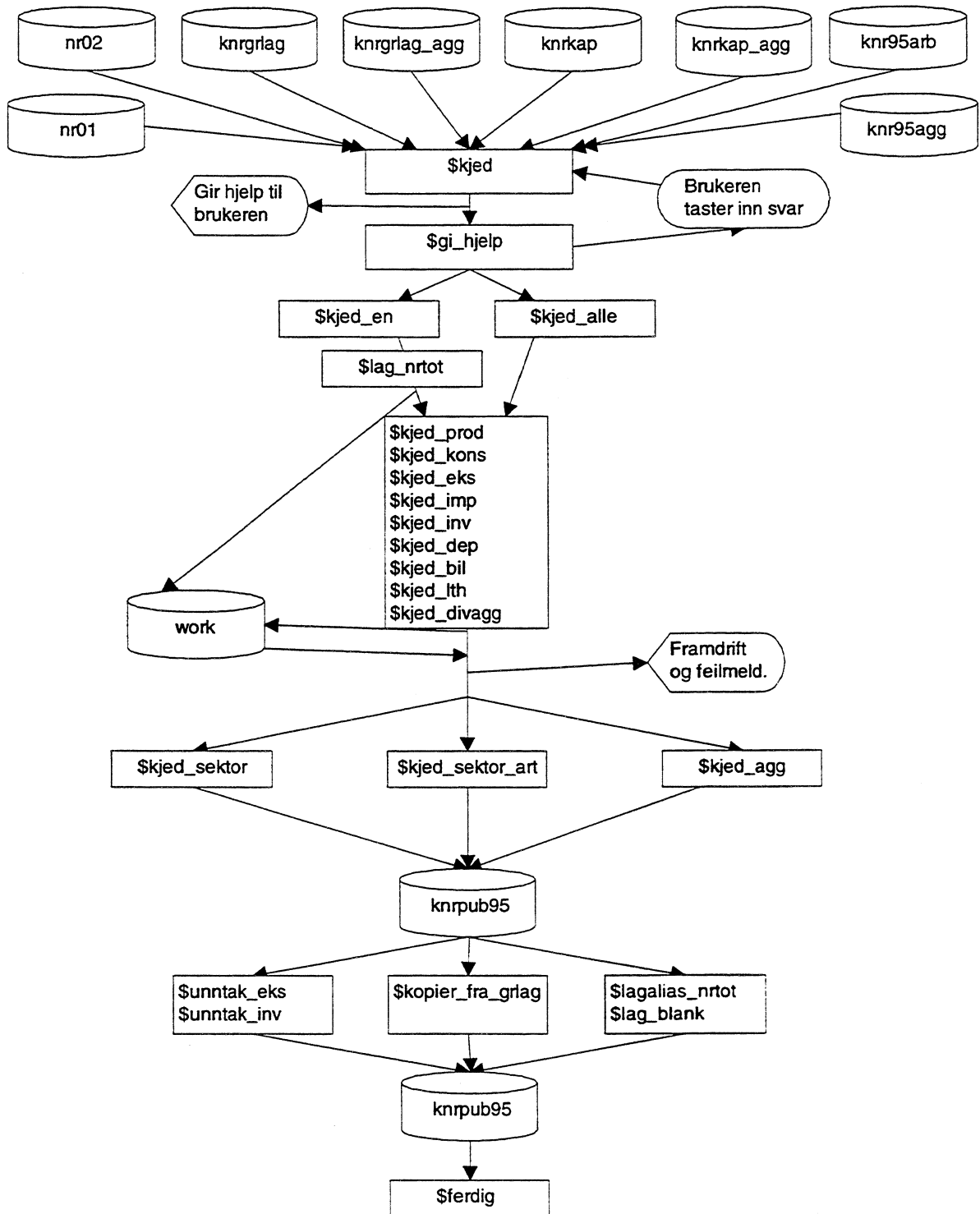
\$NASJREGN/tidsserier/NR/DB/nr01.db	«inndat» - database med t-1 priser, årstall, detaljert NR-sektor nivå (serier)
\$NASJREGN/tidsserier/NR/DB/nr02.db	«inndat» - database med t-1 priser, årstall, aggregert NR-publiseringsnivå (serier og formler)
\$NASJREGN/tidsserier/NR/DB/nrkap.db	«inndat» - database med t-1 priser, årstall (DEP, KAPN, KAPB), NR-sektor nivå (serier)
\$NASJREGN/tidsserier/NR/DB/nrkap_agg.db	«inndat» - database med t-1 priser, årstall (DEP, KAPN, KAPB), aggregert NR-publiseringsnivå (serier og formler)
\$NASJREGN/kjed/nrpub95.db	«utdat» - database med kjedingsresultater, årstall, NR-nivå (ikke alt) og aggregert NR-publiseringsnivå (6538 serier)
\$NASJREGN/tidsserier/NR/DB/knrgrlag.db	«inndat» - database med t-1 priser, årstall, KNR-sektor nivå (serier)
\$NASJREGN/tidsserier/NR/DB/knrgrlag_agg.db	«inndat» - database med t-1 priser, årstall, aggregert KNR-publiseringsnivå (formler)
\$NASJREGN/tidsserier/NR/DB/knrkap.db	«inndat» - database med t-1 priser, årstall (DEP, KAPN, KAPB), KNR-sektor nivå (serier)
\$NASJREGN/tidsserier/NR/DB/knrkap_agg.db	«inndat» - database med t-1 priser, årstall (DEP, KAPN, KAPB), aggregert KNR-publiseringsnivå (formler)
\$NASJREGN/kjed/knrpub95.db	«utdat» - database med kjedingsresultater, årstall, KNR- og aggregert KNR - publiseringnivå (11219 serier)
/ssb/frisch/fame/knr/knr.db	«inndat» - database, t-1 priser, kvartal *)
\$NASJREGN/knr/modell/knr95/knr95arb.db	kvartalsdatabase, avstemt mot årstall i knrpub95.db (5789 serier og formler)
\$NASJREGN/knr/modell/knr95/knr95korr_agg.db	kvartalsdatabase, avstemt mot årstall i knrpub95.db (791 serier), må være foran knr95agg på søkelister
\$NASJREGN/knr/modell/knr95/knr95agg.db	kvartalsdatabase (1944 aggr.formler)
\$NASJREGN/knr/struktur/kontoplan.db	lister (KNR-sektor)

*) Kvartalsdatabaser i t-1 priser og med kjedingsresultater er ikke opprettet enda

Figur 2.1. Rutiner for kjeding av databasen nrpub95.db



Figur 2.2. Rutiner for kjeding av databasen knrpub95.db*)



*) Databasene nr01 og nr02 åpnes kun pga. kjeding av serier for markedsrettet og ikke-markedsrettet virksomhet. Av samme grunn åpnes kvartalsdatabasene. Data hentes derfra for årene etter basisåret. Det har derfor ingen betydning om knr95korr_agg eller knr95agg åpnes.

Tabell 2.2 Hovedprosedyre og generelle formler for kjeding av års- og kvartalsdata

\$kjed	hovedprosedyre/styringsprogram
\$kjed_sektor	generell kjedeformel, loop gjennom 1 liste
\$kjed_agg	generell kjedeformel, loop gjennom 1 liste
\$kjed_sektor_art	generell kjedeformel, loop gjennom 2 lister

Tabell 2.3 Diverse subrutiner for kjeding av års- og kvartalsdata *)

\$kopier_fra_grlag	kopierer tall fra grunnlagsdatabasen dersom årstall på KNR-nivå kjedes (pga. avstemmingen)
\$slag_nrtot	kopierer totaler som 'nrtot' (brukes av kjedeformelprosedyrene hvis kjedeindeksen er lik null (når brukeren velger å kjede aggregatene alene).
\$slagalias_nrtot	lager alias 'nrtot', som brukes til uttak av data med valgfritt ref.år (kjøres automatisk når kjeding av hele databasen velges)

*) De øvrige subrutinene, \$sunntak_eks og \$sunntak_inv, er omtalt i avsnitt 2.5. Se også Tabell 2.5.1.

Tabell 2.4 Hjelpeprosedyrer til kjeding av års- og kvartalsdata*)

\$kjed_en	kaller generell kjedeformel for en enkelt blokk
\$kjed_alle	kaller generell kjedeformel for alle variable med et gitt aggregeringsnivå
\$gi_hjelp	skriver ut informasjon om valgmuligheter ved bruk av rutiner for kjeding og valgfritt referanseår
\$slag_blank	lager tomme serier som brukes til å sette inn blanke linjer i tabellene
\$ferdig	spør om brukeren er ferdig, og avslutter programmet, eller kaller opp \$kjed_en (evt. \$velg_refaar_en)

*) Subrutinene \$gi_hjelp og \$slag_blank og \$ferdig benyttes også av prosedyren \$velg_refaar

2.1 Styreprogrammet \$kjed

Kjedingen styres av **hovedprosedyren \$kjed**, som **kaller alle subrutinene**. De fleste beregningene foregår som tidligere nevnt ved hjelp av en generell formel (loop gjennom en «namelist»). Selve **kjedingen utføres av subrutinene \$kjed_sektor, \$kjed_sektor_art og \$kjed_agg**, som alle er ulike varianter av den generelle kjedeformelen. (Se nærmere omtale i avsnitt 2.3). Når kvartalsregnskapets basisår endres, må disse kjedeformlene rettes.

I styreprogrammet \$kjed lages dessuten formler for en del spesialaggregater til bruk ved ekstern rapportering, og til trendberegninger i KNR mv. Formlene benyttes til kjedingen, og lagres midlertidig i work-databasen til de nye seriene er opprettet, men forsvinner når man logger ut. Dersom data kjedes på aggregert KNR-nivå, kalles i tillegg prosedyren \$slag_formel. Denne formel-rutinen bør på sikt kunne utelates, og heller inngå som en del av oppdateringen av knrglag_agg.db. (Prosedyren \$slag_formel ble laget pga. avstemmingen av kvartalstallene).

Hovedprosedyren \$kjed og alle subrutinene ligger på område \$NASJREGN/kjed/, og startes ved følgende kommandoer:

load kjed

Skjed

Brukeren blir da bedt om å oppgi om kvartals- eller årstall skal kjedes, hvilket aggregeringsnivå, hvilken database resultatene skal lagres i, og om eventuelle gamle resultater skal slettes. Brukeren blir også spurt om første og siste beregningsår. Programmet tester om brukeren oppgir lovlig periodisitet, beregningsperiode osv. Videre blir brukeren spurt om kjedingen skal foretas blokkvis eller for hele databasen under ett. Deretter kaller \$kjed fortløpende diverse subrutiner som utfører kjedingen, og skriver ut meldinger på skjermen etterhvert som beregningene blir ferdige.

Hovedprosedyren \$kjed sørger for alle nødvendige lese- og skrivetillatelser. Brukeren velger selv om programmet skal åpne års- eller kvartalsdatabaser. Dersom kjeding av årsregnskapstall velges, åpnes automatisk databasene nr01, nr02, knrgrlag, knrgrlag_agg, nrkap og nrkap_agg på området \$NASJREGN/tidsserier/NR/DB/ med lesetillatelse. Dersom kjeding av årstall på KNR-nivå velges, åpnes i tillegg knrkap og knrkap_agg samt knr95arb.db, knr95korr_agg.db og knr95agg.db (kvartalsdatabasene finnes på området \$NASJREGN/knr/modell/knr95/). Kvartalsdatabasene inneholder fastpristall i 1995-priser. Alle de øvrige databasene inneholder fastpristall med t-1 som basisår. Kvartalsdatabase med årsregnskapets basisår er imidlertid ikke opprettet enda, men variabellister og loop'er er som tidligere nevnt testet. Foreløpig avsluttes programmet, og man får en feilmelding dersom man velger å kjede kvartalstall. Output-databasen med kjedingsresultatene åpnes på område \$NASJREGN/kjed/ som «utdat» (brukeren velger database-navn selv). Dessuten lastes alle prosedyrene som kalles av styreprogrammet.

Tabell 2.1.1 Variable i hovedprogrammet Skjed

Variabelnavn	Type	Lovlig verdi	Brukes til
alle_serier	namelist		navngi "blokkene", prosedyrekall
aggnivaa	string	nr/knr	velge NR- eller KNR-nivå
agg	string	u/a	velge uaggregert eller aggregert nivå
jn	string	j/n	lese inn ja eller nei
db	string	velges av bruker	åpne databasen som kjedingsresultatene skal lagres i
frekvens	string	a/q	velge periodisitet (års- eller kvartals-tall)
stap	date	årstall/årstall og kvartal	lese inn ønsket startperiode
slup	date	årstall/årstall og kvartal	lese inn ønsket sluttperiode
basisaar	date	basisåret i KNR-modellen	lage date-range ved beregning av unntak fra generell kjedeformel (unngå avvik mellom publiserte NR- og KNR-tall)
foerste_aar	date		tester om det finnes data for BNP fra og med oppgitt startperiode, stap < slup
siste_aar	date		tester om det finnes data for BNP til og med oppgitt sluttperiode, slup > stap

2.2 Blokkinnndelingen

For å bedre brukervennligheten (man slipper å «kommentere ut» deler av styreprogrammet dersom man ikke ønsker å kjede hele databanken under ett, f.eks. etter å ha foretatt mindre rettelser i input-databasen), er hovedprogrammet inndelt i blokker (se Tabell 2.3.1). Dette kan være aktuelt fordi det tar lang tid å kjøre meta-rutinene for hele databasen. Selve kjedingen av hele databanken (disaggregert KNR-nivå og KNR-aggregater) tar totalt om lag 10 minutter. Da kopieres også data fra grunnlagsdatabasen (pga. avstemmingen), og programmet gir alle totaler alias 'nrtot' (brukes til uttak av data med valgfritt referanseår, se kapittel 3).

Når hovedprogrammet \$kjed kjøres, vil brukeren få spørsmål om hele databanken skal kjedes. Hvis man svarer 'j', kalles hjelpeprosedyren \$kjed_alle. Første gang settes aggregeringskoden ('agg') lik 'u', og ved hjelp av en loop kalles alle blokk-prosedyrene, og alle serier på detaljert nivå blir automatisk kjedet. Deretter settes aggregeringskoden lik 'a'. \$kjed_alle kjøres pånytt, og kjeder alle aggregatene. Dersom aggregeringsnivå 'knr' er valgt, kopieres detaljerte serier for LOKO, LON, ARBTP, FI og DRIB fra knrgrlag pga. avstemmingen av kvartalsregnskapet. Deretter spør programmet om brukeren også vil kopiere aggregater. Til slutt får man spørsmål om programmet skal lage alias, og svarer da 'j'. Denne framgangs-måten velges normalt når databasene skal opprettes.

Tabell 2.2.1 Blokkprosedyrer for kjeding av års- og kvartalsdata

\$kjed_prod	kaller generell kjedeformel, og kjeder alle serier som følger produksjonssektorliste/produktliste
\$kjed_kons	kaller generell kjedeformel, og kjeder alle serier for konsum
\$kjed_eks	kaller generell kjedeformel, og kjeder alle serier som følger produktlister og aggregerte lister for eksport
\$kjed_imp	kaller generell kjedeformel, og kjeder alle serier som følger produktlister og aggregerte lister for import
\$kjed_inv	kaller generell kjedeformel, og kjeder alle investeringssektorer og -arter
\$kjed_lth	beregner LTH som PROB-EKS (hvis årstall), kaller generell kjedeformel, og kjeder alle serier som følger produktlister og aggregerte lister for leveranse til hjemme-markedet (benyttes til avstemming av kvartalstall)
\$kjed_dep	kaller generell kjedeformel, og kjeder alle produksjonssektorer og -arter for kapitalslit og beholdning av realkapital
\$kjed_bil	kaller generell kjedeformel, og kjeder alle serier som følger produktlister og aggregerte lister for lager og ufordelt
\$kjed_divagg	kaller generell kjedeformel, og kjeder bl.a. alle serier som inngår i tabellene makroøkonomiske hovedstørrelser og BNP etter anvendelse

Ønsker man f.eks. bare å kjede serier på disaggregert KNR-nivå, svarer man 'n' på spørsmålet om man vil kjede hele databanken. Man velger deretter aggregeringsnivå 'knr', variabel 'alle'

og aggregeringskode 'u'. Dermed kalles \$skjed_alle, og alle disaggregerte serier kjedes. I tillegg kopieres de samme disaggregerte seriene fra knrgrlag, som når hele databasen kjedes under ett. Til slutt svarer man 'j' når man blir spurt om man vil beregne unntakene fra den generelle kjedeformelen.

Alternativt kan kjedingen utføres blokkvis (man svarer 'n' på spørsmålet om man vil kjede hele data-banken), startes hjelpeprosedyren \$skjed_en. Brukeren blir bedt om å oppgi variabel og aggregerings-kode (a/u). Dersom brukeren f.eks. ønsker å kjede produksjonssektorene, og oppgir variabelen 'prod', aggregeringskode 'u', startes prosedyren \$skjed_prod. \$skjed_prod kaller fortløpende \$skjed_sektor (den generelle kjedeformelen) til PROB, PIN og BNPB er ferdig kjedet på detaljert nivå. Deretter spør hjelpeprosedyren \$skjed_en om brukeren er ferdig. Hvis brukeren svarer 'n' vil \$skjed_en kalle opp seg selv, og spørre om variabel pånytt. Hvis brukeren derimot svarer 'j', avsluttes programmet.

2.3 Generelle kjedeformler

Alle generelle kjedeformler må som tidligere nevnt rettes før databasen kan kjedes til nytt referanseår. Dette gjelder både \$skjed_sektor (se vedlegg 2), \$skjed_sektor_art og \$skjed_agg. Det kan forøvrig bli aktuelt å kjede tidsserier lenger tilbake enn 1978 dersom det beregnes nye årstall i t-1 priser. Prosedyrene må også rettes dersom man skal kjede lenger tilbake enn vi har data for pr. idag.

Subrutinen \$skjed_sektor brukes til kjeding av alle variable som følger konsumgruppe-, investeringsnæring-, investeringsart-, produksjonsnæring- eller produktlista (loop gjennom en liste). Programmet summerer først de uaggregerte seriene og beregner totalen, "nrtot", en midlertidig serie som forsvinner ved utlogging. Dersom årsprisindeksen (kjedeindeksen) er lik null et år, brukes isteden denne totalindeksen for samme variabel som kjedeindeks. Kjedingen utføres ved å multiplisere nivå-tallet i løpende priser i beregningsåret med kjedeindeksen, dvs. med en prisindeks som viser prisnivået i referanseåret i forhold til basisåret. (Dette er forøvrig forklart i publikasjonene det henvises til innledningsvis). Til slutt byttes eventuelle NC-verdier ut med null dersom samme variabel i "inndat"-databasen er lik null. Alle totaler kjedes forøvrig som tidligere nevnt på aggregert nivå.

Konsum, eksport og import kjedes etter den generelle formelen \$skjed_sektor. Som nevnt innledningsvis, blir imidlertid enkelte eksportsektorer unntatt fra den generelle kjedeformelen (se avsnitt 1.3 og 2.5).

Produksjon, bruttoprodukt, produktinnsats og kapitalslit, samt realkapital-beholdning kjedes også etter generell formel. Produksjon, bruttoprodukt og produktinnsats kjedes hver for seg i årsregnskapet, og kvartalsregnskapet avstemmes seinere mot disse seriene (gammel metode er omtalt i avsnitt 1.3).

Leveranse til hjemmemarkedet beregnes ikke i årsregnskapet, men pga. avstemmingen av kvartalsregnskapet, opprettes seriene som differansen produksjon - eksport, dersom kjeding på KNR-nivå velges. Leveranse til hjemmemarkedet kjedes forøvrig etter den generelle formelen \$skjed_sektor.

Lager kjedes etter generell formel på KNR-produktnivå. I det nåværende edb-systemet avstemmes kvartalstallene mot kjedede årstall (gammel metode er omtalt i avsnitt 1.3). I likhet med tidligere, benyttes samme produkt for (beregnet) leveranse til hjemmemarkedet som kjedeindeks. På NR-nivå er prosedyrekall for kjeding av lager foreløpig ikke lagt inn.

Residualen ufordelt lagerendring beregnes bare i kvartalsregnskapet. Seriene opprettes som følge av varebalanseringen fordi man ikke kan ha lager av tjenester, og som et resultat av modell-kjøringene. I modellen er sum ufordelt lager (u999) lik lager av produkt 990 (ds990). Ufordelt lagerendring etter KNR-produkt er lik 0 fram til basisåret. Seriene kjedes derfor ikke. Etter avstemmingen blir imidlertid avviket mellom tilgang og anvendelse i løpende priser lagt på ufordelt (del av bil.nr8lu_vr.u, lager og ufordelt totalt), men årssummen av serien skal som tidligere nevnt være lik 0.

Subrutinen **\$kjed_sektor_art** (loop gjennom to lister) er også en generell kjedeformel, og nesten identisk med **\$kjed_sektor**. Rutinen benyttes til kjeding av produksjonssektor*art (kapitalslit og realkapitalbeholdning) og investeringssektor*art (brutto- og nyinvesteringer samt salg av brukt realkapital). Til beregningene opprettes midlertidige serier som er lik null for salg av brukt realkapital dersom serier ikke finnes i «inndat»-databasen. Forøvrig blir enkelte investeringssektorer (investeringssektor*art) som nevnt innledningsvis unntatt fra den generelle kjedeformelen (se avsnitt 2.5).

Prosedyren **\$kjed_agg** er også en generell kjedeformel tilsvarende **\$kjed_sektor**, som kjeder aggregatene bl.a. til tabellene «Makroøkonomiske hovedstørrelser», «Bruttonasjonalprodukt etter anvendelse» og «Hovedtall for konsum». I motsetning til **\$kjed_sektor** brukes ikke variabelen «nrtot» (totalindeksen for samme variabel) i formelen. Dersom en kjedeindeksen likevel skulle være lik NC pga. null i nevneren, noe er lite sannsynlig, vil imidlertid programmet skrive ut en melding på skjermen.

2.4 Andre subrutiner som benyttes av hovedprogrammene

Når disaggregerte serier kjedes, beregnes først totalene i t-1 priser (utføres av de generelle kjedeformel-prosedyrene). Velger man derimot å kjede på aggregert nivå, kopierer prosedyren **\$lag_nrtot** alle totaler til work-databasen (dvs. midlertidig), og gir dem mellomnavnet «nrtot». Dette gjøres fordi programmene bruker totalen for samme variabel dersom kjedeindeksen inneholder NC-verdier. Alle «nrtot» lages forøvrig **automatisk**.

Prosedyren **\$lagalias_nrtot** gir alle totaler i databasene med kjedingsresultater et alias med mellomnavnet «nrtot». Alias benyttes i formlene for valgfritt referanseår. (Se avsnitt 3.2). Rutinen **kjøres automatisk bare dersom man velger å kjede hele databasen**.

Dersom **\$kjed_alle** kalles opp og alle variable kjedes samtidig, kjøres **Skopier_fra_grlag** automatisk. Programmet kjøres **ikke automatisk ved valg av blokkvis kjeding**. Hvilke variable som kopieres og hvorfor, er forøvrig omtalt i avsnitt 2.2.

2.5 Spesielle problemer med hensyn til metode og beregning av unntak

Som nevnt innledningsvis (avsnitt 1.3), er det likevel blitt bestemt ved seksjon 210 at enkelte variable skal behandles som unntak på samme måte som tidligere. Årsaken til at variablene som er omtalt nedenfor blir unntatt fra den generelle kjedeformelen, er at f.eks. uttaging av plattformer og prisvariasjoner på skip kan gi «rare» utslag i tallene.

Prosedyrerne **Sunntak_eks** og **Sunntak_inv** beregner unntakene først på detaljert nivå. Deretter behandles aggregatene hvor seriene inngår på samme måte. Dette **gjøres automatisk bare dersom man kjeder hele databasen under ett, fordi prosedyrene forutsetter at en del serier allerede er kjedet når unntakene skal beregnes** (**Sunntak_eks** forutsetter at PROB og SBRK er kjedet, **Sunntak_inv** forutsetter at NYI og SBRK er kjedet). Korrigerte årstall legges som tidligere nevnt i databasen knrpub95.db. Ved uttak av kvartalsdata på aggregert nivå i faste 1995-priser må \$NASJREGN/knr/modell/knr95/knr95korr_agg.db, som er avstemt mot kjedede og korrigerte årstall, være foran knr95agg.db på søkelista.

Tabell 3.5.1 Prosedyrer til beregning av unntak ved kjeding av års- og kvartalsdata

\$unntak_eks	beregner enkelte unntak for eksport (års- og kvartalstall)
\$unntak_inv	beregner unntak for enkelte investeringsarter (års- og kvartalstall)
\$skiftv *)	beregner skiftvirkninger (kvartalstall)

*) Denne prosedyren skal p.t. ikke kjøres. I stedet foretas balansering av tilgang-anvendelse i forbindelse med avstemmingen.

Prosedyreren **Sunntak_eks** korrigerer eksport etter kjeding. Eksport av borerigger og moduler til oljeplattformer blir deflatert med den ferdig kjedede indeksen for produksjon av samme produkt (352). Eksport av skip (0941) blir dessuten satt lik salg av brukt realkapital for samme art i investeringsnæringene 8305, 8365 og 8366 (fiske, innenriks og utenriks sjøfart).

Tilsvarende korrigerer **Sunntak_inv** sektor*art for bruttoinvesteringer etter kjeding. Ferdig kjedede eksporttall blir brukt til å beregning av enkelte unntak for bruttoinvesteringene. Bruttoinvestering i borerigger og moduler (0838) i næring 8311, blir satt lik nyinvesteringer minus eksport av samme investeringsart. Bruttoinvestering i fly (0842) i næring 8362 blir beregnet på tilsvarende måte. I næring 8305 blir bruttoinvestering i fiskebåter (0841) satt lik sum nyinvestering og salg av brukt realkapital for samme art. Tilsvarende blir også bruttoinvestering i skip ellers i hhv. 8365 og 8366 beregnet. Forøvrig blir de respektive aggregatene korrigert på samme måte. I det nåværende edb-systemet blir korrigerte kvartalstall summert opp pånytt, og avstemt mot kjedede årstall. Oppsummeringen foretas bl.a. ved hjelp av aggregeringsformlene.

2.6 Spesielle rutiner for beregning av fastpristall i kvartalsregnskapet

Rutinene som er omtalt i dette kapitlet, skal kun benyttes dersom man ønsker å beregne tids-serier for eksport og import lenger tilbake enn 1995, som p.t. er KNR-modellens basisår.

Med unntak av eksport og import, beregnes alle kvartalstall i løpende og faste priser ved bruk av de ordinære indikator-programmene og KNR-modellen, uavhengig av beregningsperiodens lengde. Til de vanlige, kvartalsvise indikator-kjøringene benyttes formlene i utenriks.db, som p.t. beregner prisindeksene i faste 1995-priser. Dersom man ønsker å foreta en ny tilbake-regning av kvartalsregnskapet, må formler og søkelister i prosedyrene i Tabell 2.6.1. rettes, da disse er programmert for beregning av prisindekser fra og med 1978 med 1993 = 100. Dessuten må en ny database opprettes (ur95.db med 1995 som basisår i formlene). Kvartalsfordelingsnøkklene som er omtalt i dette avsnittet, hentes forøvrig fra den gamle databasen /ssb/frisch/fame/knr/knr91.db, som kun inneholder tall tilbake til 1.kvartal 1978. Skal beregninger før 1978 utføres, må derfor alternative kvartalsfordelingsnøkler brukes, eller prosedyrene omprogrammeres.

Til kjeding av indikatorene for eksport og import benyttes en **utvidet utgave av indikator-programmet \$NASJREGN/knr/indikator/utenriks.inp**. Det opprinnelige indikator-programmet er oppdelt i et styreprogram og fire subrutiner, som alle ligger på område \$NASJREGN/kjed/. I tillegg er det laget en prosedyre for kvartalsfordeling bakover av utenriksregnskapstall (rente- og stønadsbalansetall i løpende priser).

Tabell 2.6.1. Prosedyrer

\$utenriks	styreprogram
\$slag_exim_iflg	beregner nye kvartalstall (tidsserier) ifølge HS vha. nye årsregnskapstall, og gamle kvartalstall som fordelingsnøkler (for 1978 - 1987 fordeles alt på ifølge HS, og utenom HS - seriene settes lik 0)
\$slag_exim_utenom	beregner nye kvartalstall for 1988 - 1991 utenom HS vha. nye årsregnskapstall, trekker eventuelt i fra ifølge HS, og kvartalsfordeler differansen med gamle kvartalstall utenom HS som fordelingsnøkler
\$skjed_exim_iflg	kjeder eksport og import ifølge HS
\$skjed_exim_utenom	kjeder eksport og import utenom HS
\$skjed_exim_resten	foretar oppsummering av eksport og import utenom og ifølge HS
\$slag_ur_rsbal	beregner nye kvartalstall for rente- og stønadsbalansen vha. nye årstall fra utenriksregnskapet, og gamle kvartalstall som fordelingsnøkler (lager tidsserier)

\$slag_exim_iflg beregner nye kvartalstall ifølge handelsstatistikken (HS), og lagrer resultatene i urtall.db. Prosedyren **\$slag_exim_utenom** lagrer resultatene i **exim_utenom.db** som formler. **\$skjed_exim_iflg** og **\$skjed_exim_utenom** foretar selve kjedingen. Disse kjedeformlene må som tidligere nevnt rettes (se også avsnitt 2.2 om generelle kjede-formler). Prosedyren **\$skjed_exim_resten** foretar oppsummeringen av eksport og import ifølge og utenom HS til slutt (kun mindre endringer er foretatt her). Alle prosedyrene kan forsåvidt kjøres uavhengig av selve indikator-kjøringen. Se forøvrig Teknisk dokumentasjon av beregningsopplegg for KNR (EiP), hvor indikator-kjøringene er dokumentert i kapittel 3. Man må imidlertid huske å **rette søkelista i prosedyren \$NASJREGN/knr/indikator/aapne.pro** og compilere den, før alle indikator-programmene skal kjøres. Databasen ur95.db med kjedeformlene som beregner prisindekser med 1995=100 åpnes istedenfor \$NASJREGN/knr/indikator/utenriks.db. I tillegg må også databasene **exim_utenom.db** og **\$NASJREGN/knrgml/indeks/urtall.db** åpnes.

Prosedyren \$slag_ur_rsbal benytter nye årstall fra utenriksregnskapet, og beregner tilbakegående rente- og stønadsbalansetall ved å bruke gamle tall fra /ssb/frisch/fame/knr/knr91.db som kvartalsfordelingsnøkler. Denne databasen inneholder kun tall tilbake til 1.kvartal 1978. Fra om med 1.kvartal 1992, brukes de nye, kvartalsfordelte utenriksregnskapstallene. Når årsanslagene i \$NASJREGN/knr/indeks/pia.db oppdateres bakover, må imidlertid prosedyren kjøres. Programmet \$slag_ur_rsbal kan i likhet med de øvrige rutinene for eksport/import, kjøres uavhengig av indikatorprogrammene, da tallene legges direkte i hhv. databasen på fellesområdet og i \$NASJREGN/knr/modell/knr95/knr95arb.db.

Tabell 2.6.2. Databaser

\$NASJREGN/kjed/exim_utenom.db	formler for kvartalsfordeling ev eksport/import utenom HS
\$NASJREGN/knr/indeks/urtall.db	indikatorserier utenom og ifølge handelsstatistikken
\$NASJREGN/knrgml/indeks/urtall.db	gamle indikatorserier utenom og ifølge handelsstatistikken
\$NASJREGN/kjed/ur95.db	indikator-formler for tilbakeregning (må evt. opprettes)
\$NASJREGN/knr/indeks/pia.db	manuelt oppdaterte indikatorserier, årsanslagene som benyttes av \$slag_ur_rsbal til kvartalsfordeling av rente- og stønadsbalansetall

2.7 Kontroll av resultatene

Etter at årstall er kjedet på KNR-nivå, må man dessuten **sjekke at alle serier som finnes i kvartalsdatabasene, også finnes i databasen med kjedingsresultater**. Dette kan gjøres vha. input-fila \$NASJREGN/kjed/sammenlign.inp, som sjekker at års- og kvartalsdatabasene inneholder de samme seriene. Hvis noen dataserier ikke finnes i knrpub95.db, vil avstemmingsprogrammene stoppe opp hver gang en serie mangler. Avstemmingsrutinene bør ikke endres slik at de likevel fortsetter dersom serier mangler, da det ville ha medført at disse kvartals-seriene ikke ville blitt avstemt. Kjedeformel-prosedyrene skriver som tidligere nevnt ut melding både på skjermen og til fila \$NASJREGN/kjed/feilmelding.txt, dersom enkelte aggregater inneholder NC-verdier, eller mangler i input-databasen(e). I såfall må formelen lages, og den blokk variabelen inngår i, må kjøres pånytt. Dersom en serie på detaljert nivå ikke finnes i årsregnskaps-databasen, settes den imidlertid lik null.

Til slutt bør vekstrater kontrolleres (f.eks. ved å sjekke tabellen med **Makroøkonomiske hovedstørrelser**) mot tidligere publiserte tall dersom kjedeformler er endret (se avsnitt 2.7 om teksting av variable og uttak av tabeller).

På området /ssb/frisch/fame/knr/korriger/ finnes dessuten diverse **kontroll-rutiner** til å sjekke tilgang - anvendelse, at alt er avstemt i kvartalsdatabasene mv.

2.8 Når databanken er ferdig kjedet

Etter at seriene er ferdig kjedet, bør man legge inn variabel-beskrivelser (description) til alle seriene i databasen. Det er en stor ulempe å måtte legge inn beskrivelser hver gang kjedingsprogrammene kjøres, da det tar lang tid å kjøre metarutinene (dvs. programmene som legger

inn tekstene). Det hadde derfor vært tidsbesparende om man kunne benyttet formler istedenfor serier. På den annen side inneholder databasene svært mange variable, og det utføres også en del mellomberegninger underveis. Dessuten MÅ de generelle kjedingsformlene oppdateres før kjeding. I tillegg MÅ data på KNR-nivå opprettes som serier for å kunne leses av TROLL-programmene i forbindelse med avstemmingen (TROLL leser ikke formler).

Det hadde også vært tidsbesparende om man kunne utelatt beskrivelsene. Beskrivelser må imidlertid være lagt inn dersom det gamle tabellprogrammet skal brukes til å ta ut tabeller. De nye tabellrutinene krever derimot ikke at det er lagt inn beskrivelser (de hentes fra tekstfiler hver gang programmet kjøres), men det er mye enklere for brukerne å benytte seg av databasene når alle tidsserier er dokumentert. Det kan derfor være lønnsomt å kjøre beskrivelsesrutinene årlig likevel.

Prosedyren knrmeta.pro (Eip) brukes til teksting av kvartalsvariable på KNR-nivå og KNR-publiseringsnivå. Prosedyren ligger på \$NASJREGN/knr/struktur/. Det er dessuten laget tilsvarende prosedyrer til teksting på ulike aggregeringsnivåer til årsregnskapet. Disse programmene (EiP) ligger på område \$NASJREGN/tidsserier/meta/. Se forøvrig Teknisk Dokumentasjon av realregnskapets FAME-databaser (EiP 1996, STo og PET 1998).

Bruk av tabellprogrammet for uttak av kjedede data er nærmere omtalt i avsnitt 4.

3 Valg av nytt referanseår

Ved eksterne henvendelser blir vi ofte forespurt om data med et annet referanseår enn man har for de publiserte tallene. Dette gjelder såvel norske brukere som internasjonale organisasjoner, først og fremst OECD, IMF og EUROSTAT. Det er derfor laget rutiner som gjør det mulig å for brukeren å velge referanseår innenfor den perioden vi har nasjonalregnskapstall for pr. idag. Prosedyrene lager **formler**, og man vil automatisk få med rettelsler i tallserier hver gang formlene eksekveres, noe som innebærer **minimalt behov for vedlikehold og oppdatering**. Bruk av formlene forutsetter normalt kun at databanken med KNR-basisårets priser er ferdig kjedet. Formlene, som kan brukes til uttak av både års- og kvartalstall, er lagret i en egne databaser, henholdsvis `velg_refaar_nrpub`, `velg_refaar_nr` og `velg_refaar_knr` (se Tabell 3.2).

Tabell 3.1 Hoved- og hjelpeprosedyrer for valgfritt referanseår *)

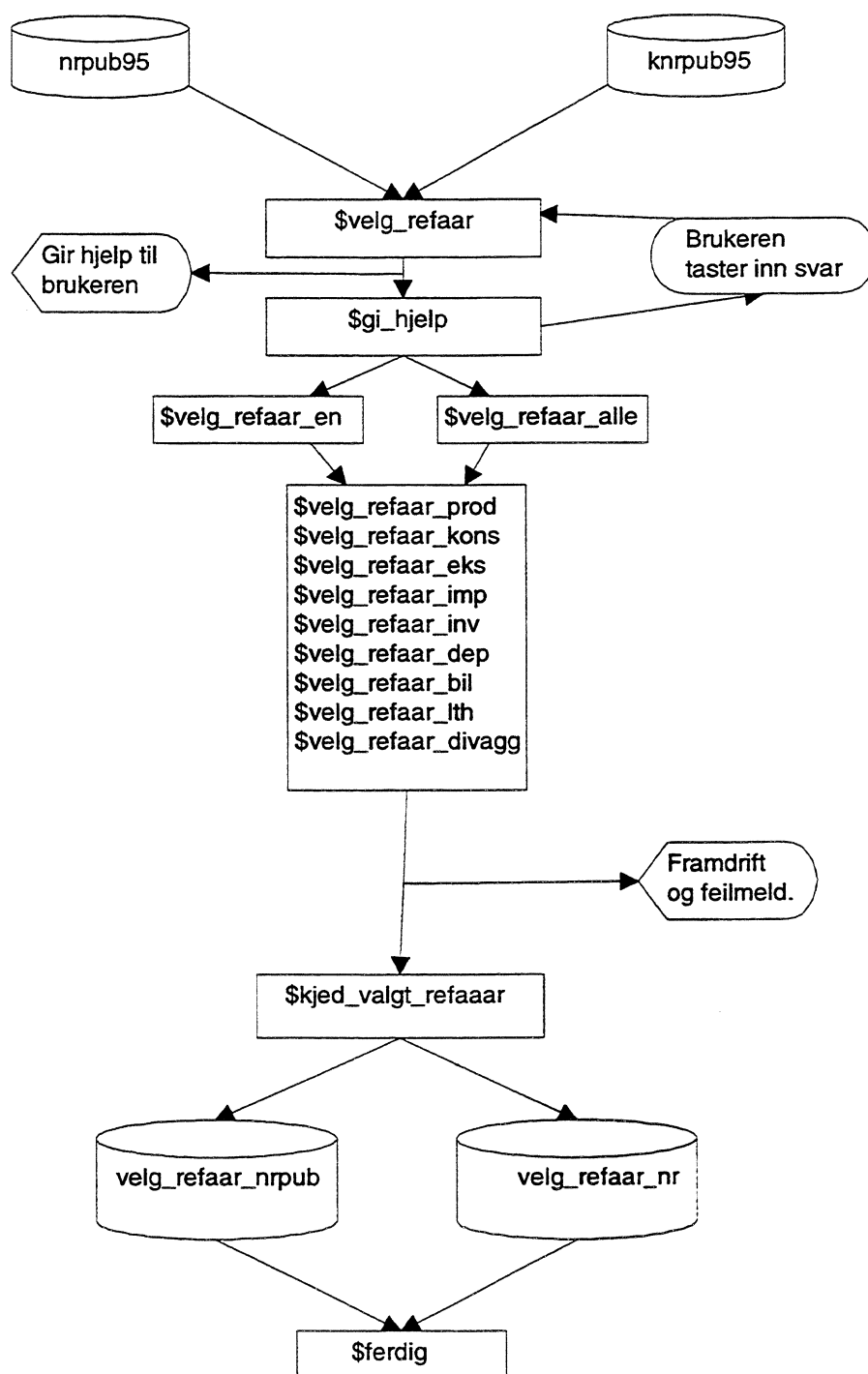
<code>\$velg_refaar</code>	hovedprosedyre/styringsprogram (kan kjøres uavhengig av de øvrige kjedings-rutinene, men har enkelte hjelpe-prosedyrer felles med <code>\$kjed</code>)
<code>\$velg_refaar_en</code>	kaller prosedyren <code>\$kjed_valgt_refaar</code> , som lager formlene
<code>\$velg_refaar_alle</code>	kaller prosedyren <code>\$kjed_valgt_refaar</code> , som lager formlene
<code>\$kjed_valgt_refaar</code>	lager formlene som utfører beregningene
<code>\$gi_hjelp</code>	skriver ut hjelp på skjermen
<code>\$lag_blank</code>	lager tomme serier som brukes til å sette inn blanke linjer i tabellene
<code>\$ferdig</code>	spør om brukeren er ferdig, kaller evt. <code>\$velg_refaar_en</code> eller <code>\$gi_hjelp</code>

*) Subrutinene `$gi_hjelp`, `$lag_blank` og `$ferdig` brukes også av prosedyren `$kjed`

Tabell 3.2 Databaser

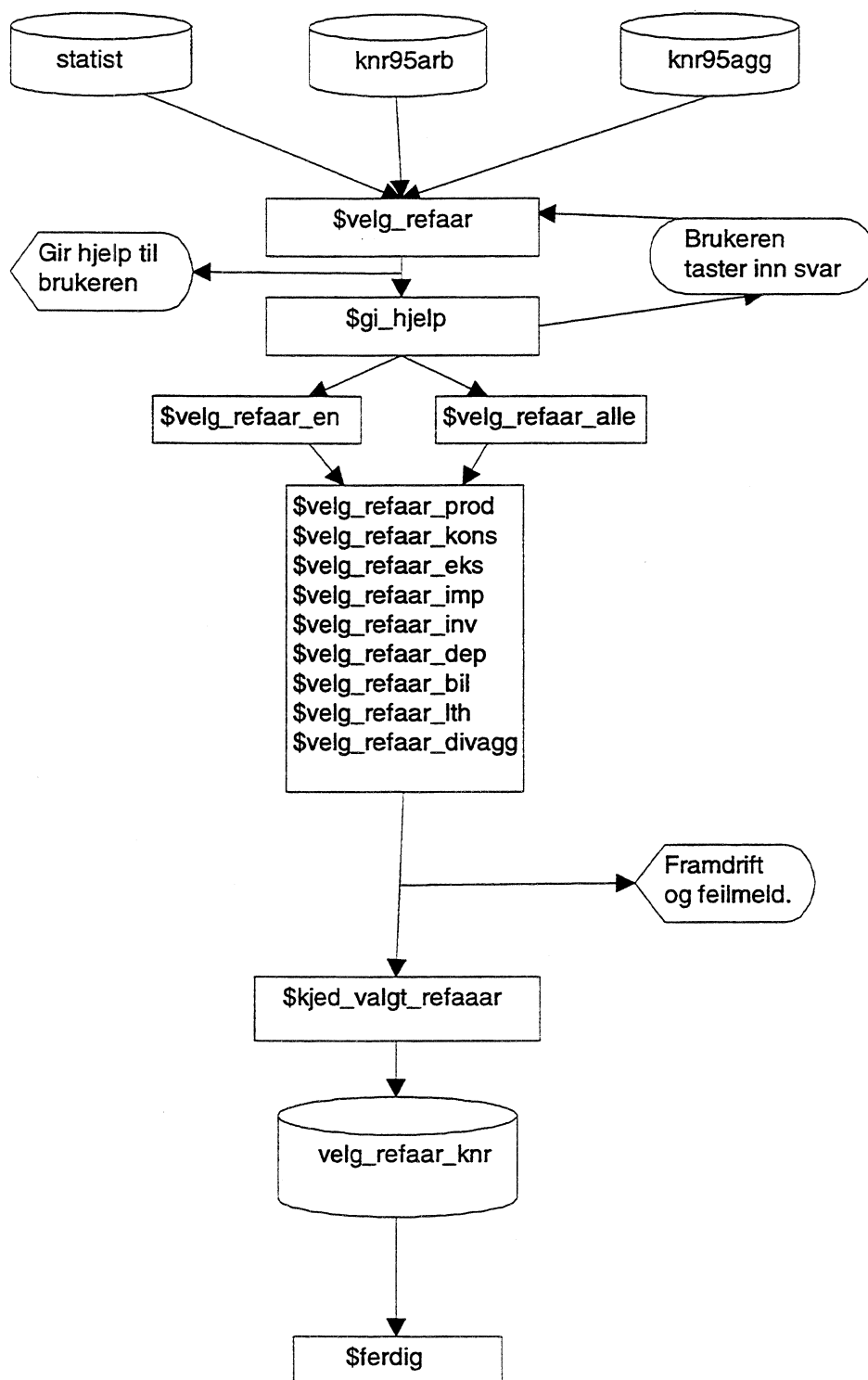
<code>\$NASJREGN/kjed/velg_refaar_nrpub.db</code>	formeldatabase (2610), årstall på NR- og nrpub-nivå
<code>\$NASJREGN/kjed/velg_refaar_nr.db</code>	formeldatabase (5066), årstall på KNR-sektor og KNR-publiseringsnivå
<code>\$NASJREGN/kjed/velg_refaar_knr.db</code>	formeldatabase (3314), kvartalstall på KNR-sektor og KNR-publiseringsnivå
<code>\$NASJREGN/knr/modell/knr95/knr95arb.db</code>	1995-priser, kvartal, avstemt mot årlig regnskap
<code>\$NASJREGN/knr/modell/knr95/knr95korr_agg.db</code>	1995-priser, kvartal, avstemt mot årlig regnskap på aggregert nivå (volumtall)
<code>\$NASJREGN/knr/modell/knr95/knr95agg.db</code>	aggregeringsformler, KNR
<code>\$NASJREGN/kjed/nrpub95.db</code>	kjedingsresultater, 1995-priser, årstall
<code>\$NASJREGN/kjed/krpub95.db</code>	kjedingsresultater, 1995-priser, årstall
<code>\$NASJREGN/knr/struktur/kontoplan.db</code>	lister (knr-sektor)

Figur 3.1. Rutiner for å lage omregningsformler, NR- og KNR-nivå, årstall*)



*) \$velg_refaar_lth kjøres bare dersom KNR-nivå velges

Figur 3.2. Rutiner for å lage omregningsformler, kvartalstall*)



*) Databasen statist åpnes etter knr95arb og knr95korr_agg, og kun for å lage variabelliste for hele databasen. Man vil derfor likevel få ut de riktige vekstratene for aggregatene når \$velg_refaar åpner alle databasene

3.1 Styreprogram

I likhet med de øvrige rutinene for kjeding av nasjonalregnskapstall, ligger styreprogrammet \$velg_refaar og subrutinene (noen av dem benyttes også av prosedyren \$kjed, se Tabell 3.1) på område \$NASJREGN/kjed/. Programmet startes med kommandoen

```
load velg_refaar
$velg_refaar
```

Styreprogrammet \$velg_refaar åpner databasene med kjedede tall og lister. Listene som følger KNR-nivå, hentes fra kontoplan.db på område \$NASJREGN/knr/struktur/. Listene på aggregert nivå lages på tilsvarende måte som beskrevet i avsnitt 3. Prosedyren \$velg_refaar er forøvrig bygget opp som \$kjed (blokkindelning), og man kan om ønskelig legge inn flere prosedyrekall på nesten tilsvarende måte (hvilke prosedyrekall som allerede er lagt inn framgår av programkoden for \$velg_refaar, se Vedlegg 3).

Tabell 3.1.1 Variable i hovedprogrammet \$velg_refaar

Variabelnavn	Type	Lovlig verdi	Brukes til
alle_serier	namelist		navngi "blokkene", prosedyrekall
hele_db	namelist		loop gjennom hele databasen
ingen_liste	namelist		tom liste (hjelpevariabel)
alle_nrpub	case - serie		lage variabelliste for loop gjennom hele db
alle_nr	case - serie		lage variabelliste for loop gjennom hele db
alle_knr	case - serie		lage variabelliste for loop gjennom hele db
aggnivaa	string	nr/knr	velge nr- eller knr-nivå
agg	string	u/a	velge uaggregert eller aggregert nivå
jn	string	j/n	lese inn ja eller nei
db	string	bestemmes når brukeren har valgt frekvens, (evt) aggnivaa	åpne databasen som formlene skal lagres i for skriving eller gi lesetillatelse, enten velg_refaar_nrpub, velg_refaar_nr eller velg_refaar_knr
frekvens	string	a/q	velge periodisitet (års- eller kvartals-tall)
referanse_aar	date	årstall	lese inn ønsket referanseår
foerste_aar	date		bestemme dato-range, tester om brukeren har valgt gyldig referanseår, dvs. om serien for BNP er langt nok oppdatert bakover
siste_aar	date		bestemme dato-range, tester om brukeren har valgt gyldig referanseår, dvs. om serien for BNP er langt nok oppdatert framover

Programmet spør brukeren om hvilken periodisitet og aggregeringsnivå formlene skal ha, og om brukeren skal åpne databasene for å lage formler eller innhente lesetillatelse. Programmet åpner automatisk databaser med grunnlagsårets priser når periodisitet og aggregeringsnivå er valgt. Beregningsperioden defineres automatisk ut i fra hvor langt BNP er oppdatert i de

respektive databasene. Videre spør programmet om hvilket år som skal være referanseår, og tester deretter om brukeren har valgt gyldig referanseår. Referanseåret må være innenfor det intervallet som BNP er oppdatert for. Dersom referanseåret brukeren har valgt er mindre enn 1978 eller større enn 1997 (1995 på nrpub-nivå), som vi p.t. har data for, blir det skrevet ut en feilmelding, og brukeren blir bedt om å oppgi referanseår pånytt. Ønsker brukeren for eksempel bare å se på enkelte serier, eller kjøre ut en tekstfil for import til regneark, avsluttes styreprogrammet. Hvis brukeren derimot ønsker å oppdatere formlene, kalles subrutinen \$gi_hjelp, som skriver ut hjelp på skjermen. Deretter kalles prosedyren \$velg_refaar_en eller \$velg_refaar_alle inntil brukeren velger å avslutte. Det er imidlertid bare nødvendig å kjøre subrutinene dersom kontoplanen eller aggregatene endres, formlene rettes eller antall variable økes.

3.2 Generelle omregningsformler

Proseduren \$velg_refaar gir brukeren valg mellom å oppdatere alle variablene samtidig eller enkeltvis. Årsaken til at denne valgmuligheten er lagt inn, er at man da slipper å kjøre de tidkrevende beskrivelsesprosedyrene for hele databanken, dersom det kun er foretatt noen få endringer (enten nye serier eller aggregeringsformler). Når brukeren har valgt variabel (variable) og aggregeringsnivå, kalles prosedyren \$kjed_valgt_refaar fortløpende. Dersom man ønsker å lage formler for serier eller aggregater som følger nye lister, må man forøvrig legge inn prosedyrekall i de respektive blokkene, og kompilere \$velg_refaar. Dersom serier er rettet (dvs. ikke nyopprettet), er det imidlertid ikke nødvendig å lage nye formler.

Formlene lages av prosedyren \$kjed_valgt_refaar. Nye volumtall får etternavn «vlr». Forøvrig er vanlig level1-navnestandard fulgt. I formlene benyttes volumindekser til beregning av nye fastpristall. Den nye serien blir eventuelt satt lik null, dersom verditallet er lik null. Hvis verditallet derimot er forskjellig fra null, beregnes volumserie med nytt referanseår. Dersom volumindeksen for en variabel er lik null, brukes isteden totalindeksen for samme variabel. Omregning for års- og kvartalstall foregår på samme måte, men når kvartalstallene beregnes, tas det hensyn til at seriene som inngår i uttrykket har forskjellig periodisitet. Eksempler på omregningsformler er vist nedenfor.

Eksempel 1: Omregning av kjedede årstall for KNR-næring 2305 til nytt referanseår

VELG_REFAAR_NR'BNPB.NR2305.VLR Bruttoprodukt, basisverdi_Fiske og fangst_Volum (rebasert)

```
IF BNPB.NR2305.VL EQ 0 THEN 0 ELSE BNPB.NR2305.VL*
(IF BNPB.NR2305.VR[REFERANSE_AAR] EQ 0 THEN
BNPB.NRTOT.VR[REFERANSE_AAR]/
BNPB.NRTOT.VL[REFERANSE_AAR] ELSE
BNPB.NR2305.VR[REFERANSE_AAR]/BNPB.NR2305.VL[REFERANSE_AAR])
```


Eksempel 2: Omregning av kvartalstall for KNR-sektor 2305 i faste priser til nytt referanseår

VELG_REFAAR_KNR'BNPB.NR2305.VLR.U Bruttoprodukt, basisverdi_Fiske og fangst_Volum (rebasert)_Ujustert

```
IF BNPB.NR2305.VL.U EQ 0 THEN 0 ELSE BNPB.NR2305.VL.U*
(IF BNPB.NR2305.VR[REFERANSE_AAR] EQ 0 THEN
BNPB.NRTOT.VR[REFERANSE_AAR]/4/
AVE((BNPB.NRTOT.VL/4)<DATE REFERANSE_AAR>)
ELSE BNPB.NR2305.VR[REFERANSE_AAR]/4/
AVE((BNPB.NR2305.VL/4)<DATE REFERANSE_AAR>)
```

Subrutinen \$ferdig avslutter programmet dersom brukeren er ferdig med å oppdatere formler. I motsatt fall kalles \$velg_refaar_en igjen, eventuelt \$gi_hjelp etter brukerens ønske.

3.3 Uttak av data med nytt referanseår

Man behøver ikke å legge inn variabel-beskrivelser om man ønsker å bruke (det gamle) tabellprogrammet, med mindre formel-databasen er oppdatert (se avsnitt 3.1). Dersom endringer er foretatt, må variablene tekstes som beskrevet i avsnitt 2.8. Generelt må man imidlertid være påpasselig mht. søkeliste (se del 4).

4 Generelt om uttak av data

Avslutningsvis gjennomgås uttak av nasjonalregnskapstall generelt. Fremstillingen vil imidlertid bli kortfattet, da bruk av tabellprogrammene er dokumentert andre steder (Cal 1995, EiP 1996, NAL 1998). **Man må forøvrig være oppmerksom på at alle databasene (unntatt formeldata-basene) som er omtalt i dette notatet, vil inneholde uoffisielle tall mens det arbeides med de forskjellige regnskapene. Ved å følge “oppskriften” nedenfor, får man altså ut “riktige”, men ikke nødvendigvis offisielle tallserier.** Hvilke databaser som til enhver tid inneholder offisielle tall, er også dokumentert andre steder (NAL 1998).

Tidligere hadde man ikke noe opplegg for å skrive ut enkeltserier med nytt referanseår på skjermen eller på papir, eventuelt til tekstfil eller til regneark. Da man nå har formler for omregning til nytt referanseår, kan brukerne av seksjonens data enkelt få regnearkfiler eller databaser med tidsserier som har ønsket basisår (p.t. $1978 \leq 1997$). Skal man skrive ut en tidsserie, f.eks. BNP med 1990 = 100, åpner man først databasene, og velger referanseår og aggregerings-nivå slik (programmet vil stille spørsmål):

```
load velg_refaar
$velg_refaar
```

Deretter gis f.eks. følgende kommandoer fra dialog-vinduet i FAME:

```
output <acc over> bnp.txt
disp bnpb.nr23_9.vlr
output terminal
```

Fila bnp.txt kan f.eks. enten skrives ut på printeren, overføres til PC eller sendes via e-mail.

For å lage tekstfiler for overføring til EXCEL, kan eventuelt prosedyren \$fame2exc (FAME-kontoret i London) på område \$NASJREGN/knr/tabell/ benyttes. Prosedyrene \$oecd2excel (PET), som kaller \$fame2exc, og imf2excel (PET), benyttes til å kjøre ut data til hhv. OECD og IMF i faste 1990- og 1992-priser. Begge skriver ut data hhv. vertikalt og horisontalt etter brukernes ønske. Disse prosedyrene finnes på område \$NASJREGN/knr/tabell/internasjonalt/.

Som tidligere nevnt, må alle variable ha beskrivelser dersom man ønsker å ta ut tabeller vha. det gamle tabellprogrammet. Før man kan ta ut tabeller, må man også ha riktig søkeliste i tabellprogrammet. Ofte vil rekkefølgen databasen åpnes i være viktig, dersom flere databaser inneholder variable med samme navn, eller dersom alle databasene ikke er oppdatert (ved **ekstern rapportering må man naturligvis også passe på at ikke uoffisielle tall sendes ut**).

For å ta ut kvartalstall fra **arbeidsdatabasene** i 1995-priser, **MÅ** databasene p.t. åpnes i denne rekkefølgen:

```
open <access read> "$NASJREGN/knr/modell/knr95/knr95arb" as knr95arb
open <access read> "$NASJREGN/knr/modell/knr95/knr95korr_agg" as knr95korr_agg
open <access read> "$NASJREGN/knr/modell/knr95/knr95agg" as knr95agg
```

Dersom man ønsker å ta ut kvartalstall med et annet referanseår enn 1995, må variabelen referanse_aar, som brukes i formlene, være definert i input-fila eller tabellprogrammet. Dessuten må man i tillegg åpne databasen(e) med **omregningsformlene**. Om disse **åpnes før eller etter de ovennevnte databasene har ingen betydning**, da variablene har forskjellig navn.

```
open <access read> "$NASJREGN/kjed/velg_refaar_knr" as vknr
open <access read> "$NASJREGN/kjed/knrpub95" as nr
```

MEN: knr95korr_agg.db må **ALLTID** åpnes før knr95agg.db dersom man skal ta ut tilbakegående kvartalstall i pris eller volum (dvs. fra og med 1978 til og med KNR-basisåret, p.t. 1995) fra **arbeidsdatabasene**.

De **“offisielle”** utgavene av knr95korr_agg.db oppdateres etter siste KNR-kjøring, da avstemte aggregater overføres til \$REFERTID/data/statist.db (alt lagres som serier) og til statist-databasen på fellesområdet /ssb/frisch/fame/knr/. Innholdet i aggregeringsformlene framgår av de med “etternavn” VR.U (verdi) i den sistnevnte databasen. **Åpningsrekkefølgen** for disse databasene har **ingen betydning**.

Ved frigivelse av nye årstall blir både løpende priser, t-1 priser og kjedede tall overført til hhv. \$NASJREGN/tidsserier/NR/DB/nr.db og \$NASJREGN/tidsserier/NR/DB/knraar.db. Ønsker man å ta ut data med et annet referanseår, åpnes i tillegg databasene med omregningsformler. **Åpningsrekkefølgen** for disse databasene har heller **ingen betydning**.

På områdene \$NASJREGN/tidsserier/tabell/ og \$NASJREGN/knr/tabell/ finnes det forøvrig diverse input-filer og prosedyrer til uttak av tabeller. Dersom man ønsker å ta ut data fra andre områder enn de vanlige databasene, må man rette i tabell-prosedyren og kompilere den før tabellene kjøres ut. En annen mulighet er å legge inn større valgfrihet med hensyn til databaser i tabellprogrammet, da man ofte vil ønske å kjøre ut data fra forskjellige områder.

Tabell 5.1 Tabellpakker for uttak av kjedede årstall og serier med valgfritt referanseår

\$NASJREGN/tidsserier/tabell/knr_kjed_uagg.inp	skriver ut årstall, disaggregert KNR-nivå
\$NASJREGN/tidsserier/tabell/knr_kjed_agg.inp	skriver ut årstall, KNR publiseringsnivå
\$NASJREGN/tidsserier/tabell/knr_vrefaar_uagg.inp	skriver ut årstall, disaggregert KNR-nivå
\$NASJREGN/tidsserier/tabell/knr_vrefaar_agg.inp	skriver ut årstall, KNR publiseringsnivå
\$NASJREGN/knr/tabell/knr_vrefaar.pro	skriver ut kvartalstall, KNR publ.nivå

For å kunne velge referanseår direkte når man tar ut tabeller på samme måte som tidligere, må også spørsmål om hvilket referanseår brukeren ønsker legges inn i tabellprogrammets hovedmeny. Det bør også testes på om valg av referanseår er gyldig. Dersom man bruker styreprogrammet \$velg_refaar til å åpne databaser, bestemme beregningsperiode og referanseår, sjekker programmet først om BNP er oppdatert innenfor den perioden brukeren vil ha som referanseår. Hvis ikke serien er lang nok, ber programmet brukeren om å oppgi referanseår på nytt. Tilsvarende tester kan legges inn i tabellprogrammene.

Det er dermed mulig å forbedre brukerens valgfrihet, uten å foreta større endringer i tabellrutinene.

5 Referanser

NOS Nasjonalregnskapsstatistikk 1991

NOS Kvartalsvis Nasjonalregnskap 1978 - 1990

Teknisk dokumentasjon av beregningsopplegg for KNR (Eirin Pettersen 1996, Steinar Todsén og Pia E. Tønjum 1998)

Teknisk dokumentasjon av realregnskapets FAME-databaser (Eirin Pettersen 1996, Steinar Todsén og Pia E. Tønjum 1998)

FAME og nasjonalregnskapet. Dataflyt og databaser. Upublisert notat. (Nils Amdal 1998)

Teknisk dokumentasjon av beregningsopplegget for det kvartalsvise nasjonalregnskapet. Interne Notater 85/15 (Hilde Olsen og Arent Skjæveland 1995)

Kvartalsvis nasjonalregnskap - dokumentasjon av beregningsopplegget (Steinar Todsén m.fl. 1998).

Avstemming av kvartalsvise nasjonalregnskapsdata mot årlige nasjonalregnskap. Interne Notater 85/22 (Arent Skjæveland 1995)

Rutiner for avstemming av kvartalstall mot årstall. Upublisert notat. (Jørgen Ouren 1995)

Litt om avstemming av kvartalstall mot årstall. Upublisert notat. (Jørgen Ouren 1996)

Dokumentasjon av tabellprogram for publisering av NR- og KNR-tabeller. Upublisert notat. (Alnæs Cecilie 1995)

Standard tabellapplikasjon til uttak av FAME-data. Upublisert notat. (Nils Amdal og Erik Sjøberg 1998)

UNIX Kurs- og brukerhefte. Interne dokumenter 95/10. (Erik Sjøberg 1995)

Innføring i FAME. Interne notater 94/18. (Yngve Vogt 1994)

Statistisk sentralbyrås håndbøker 63: Håndbok i IT-metode, Versjon 2.0

6 Vedlegg

Vedlegg 1: Styreprogrammet \$kjed

```
-- Programmert av Pia E. Tønjum, 95/96
-- Sist endret: PET,25/10-96
-- Rutiner for kjeding av årstall i t-1 priser på knr- og/eller nrpub-nivå,
-- disaggregert/aggregert (evt. kvartalstall dersom database med t-1 priser
-- opprettes) til samme basisår som i KNR-modellen. Kjedeformler må rettes
-- når basisåret endres før programmene kan kjøres. Kjedingen kan utføres
-- enten for hele databanken under ett eller blokkvis
-- Lagt inn basene til tilbakeregningen NAL lør 30/11 96
```

```
PROCEDURE $kjed                                -- styrer kjedingen
```

```
block
clear
close all
over on
ignore addition on
ignore function on
ignore multiplication on
glue dot
channel warning none
date * to *
```

```
-- LASTER PROSEDYRENE
```

```
load "$NASJREGN/kjed/kjed_agg"                -- loop gjennom 1 liste
load "$NASJREGN/kjed/kjed_sektor"            -- loop gjennom 1 liste
load "$NASJREGN/kjed/kjed_sektor_art"        -- loop gjennom 2 lister
load "$NASJREGN/kjed/lag_formler"            -- lager div. aggregater til KNR
load "$NASJREGN/kjed/kopier_fra_grlag"       -- (pga. avstemming mot nr)
load "$NASJREGN/kjed/lag_nrtot"              -- hvis kun agg-nivå kjedes
load "$NASJREGN/kjed/lagalias_nrtot"         -- (brukes av prosedyren
-- kjed_valgt_refaar)
```

```
-- I utg.pkt. kun for Forskningsavd, nå likt
load "$NASJREGN/kjed/unntak_eks"             -- div. unntak, eksport
load "$NASJREGN/kjed/unntak_inv"            -- div.unntak, inv.sektor
```

```
-- kjøres normalt ikke (kvartal)
load "$NASJREGN/kjed/skiftv"                 -- skiftvirkninger
```

```
-- hvis omregningsformler skal oppdateres
load "$NASJREGN/kjed/velg_refaar"           -- evt. valg av referanseår
```

```
-- liste med alle blokkene som inngår i kjedingsrutinene
-alle_serier = {prod,bil,kons,inv,eks,imp,lth,dep,divagg}
```

```
-- SKRIVER UT INFO/HJELP TIL BRUKEREN
```

```
type blank
type "-----"
type "HUSK Å RETTE KJEDE-FORMLER HVIS NYTT GRUNNLAG/BASISÅR!"
type "(Programmet avbrytes med Ctrl + c)"
```

```

type "-----"
type blank
type blank

```

```
$gi_hjelp
```

```
-----
-- ÅRS- ELLER KVARTALSTALL/PERIODISITET
-----
```

```

response keyboard
scalar work'frekvens: string
accept "Kvartals- eller årstall (q/a)? ":frekvens

```

```

if frekvens eq "q"
  freq q
else if frekvens eq "a"
  freq a
else
  type "Ugyldig periodisitet: " + string(frekvens)
  accept "Års- eller kvartalstall (a/q)? ":frekvens
end if

```

```
-----
-- AGGREGERINGSNIVÅ (KNR ELLER NR)
-----
```

```

scalar work'aggnivaa: string
scalar work'agg: string

```

```

if frekvens eq "a"
  accept "Årstall etter KNR- eller NR-nivå (knr/nr)?" :aggnivaa
  if not (aggnivaa eq "knr" or aggnivaa eq "KNR")
    if not (aggnivaa eq "nr" or aggnivaa eq "NR")
      type "Ugyldig aggregeringsnivå!"
      accept "Årstall etter KNR- eller NR-nivå (knr/nr)?" :aggnivaa
    end if
  end if
else
  -/aggnivaa = "knr"
end if

```

```
-----
-- LESETILLATELSER
-----
```

```
open <acce r> "$NASJREGN/knr/struktur/kontoplan" as kplan
```

```

if frekvens eq "q"
  type "Database med t-1 som basisår ikke opprettet enda!"
  return
else

```

```

open <acc r> "$NASJREGN/tidsserier/NR/DB/knrglag" as knrglag
open <acc r> "$NASJREGN/tidsserier/NR/DB/knrglag_agg" as knrglag_agg

```

```

if aggnivaa eq "knr"
  open <acc r> "$NASJREGN/tidsserier/NR/DB/knrkap" as knrkap
  open <acc r> "$NASJREGN/tidsserier/NR/DB/knrkap_agg" as knrkap_agg
  open <acc r> "$NASJREGN/knr/modell/knr95/knr95arb" as knr95arb
  open <acc r> "$NASJREGN/knr/modell/knr95/knr95korr_agg" as knr95korr_agg
  open <acc r> "$NASJREGN/knr/modell/knr95/knr95agg" as knr95agg
  open <acc r> "$NASJREGN/knr/indeks/pia" as pia
end if

```

```

open <acc r> "$NASJREGN/tidsserier/NR/DB/nr01" as nr01
open <acc r> "$NASJREGN/tidsserier/NR/DB/nr02" as nr02
open <acc r> "$NASJREGN/tidsserier/NR/DB/nrkap" as nrkap
open <acc r> "$NASJREGN/tidsserier/NR/DB/nrkap_agg" as nrkap_agg

```

```
end if
```

```

type "Følgende databaser er åpnet med lesetillatelse: " + @open.db
type blank

```

```
try; open <acc over; kind text> feilmelding; end try
```

```
-- OPPRETTER NY, ÅPNER ELLER TØMMER EKSISTERENDE DB
```

```

scalar work'db:string
scalar work'jn:string
accept "Hvilken db skal de kjedede tallene lagres i? ":db
accept "Skal alle gamle resultater slettes (j/n)? ":jn

if jn eq "j"
  open <access over> file(db) as utdat
  type "Databasen " +name(id(db))+ " overskrives, og åpnes som UTDAT"
else if jn eq "n"
  try
    open <access shared> file(db) as utdat
    type "Har åpnet den eksisterende databasen "+name(id(db))+ " som UTDAT"
  otherwise
    open <access create> file(db) as utdat
    type "Har åpnet den nye databasen " +name(id(db))+ " som UTDAT"
  end try
end if

store utdat

```

```
-- VALG AV BEREGNINGSPERIODE/TESTER OM DATA FINNES INNENFOR
-- ØNSKET INTERVALL
```

```

scalar work'basisaar:date(annual) = 1995          -- brukes til dato-range
scalar work'stap:date(annual)
scalar work'slup:date(annual)
accept "Første beregningsår? ":stap
accept "Siste beregningsår? ":slup

scalar <store work> foerste_aar: date = &&
      (if frekvens eq "a" and aggnivaa eq "knr" then &&
        firstvalue(bnpb.nr23_9.vr) &&
        else if frekvens eq "a" and aggnivaa eq "nr" then &&
        firstvalue(bnpb.nr2_.vr) &&
        else firstvalue(bnpb.nr23_9.vr.u))
scalar <store work> siste_aar: date = &&
      (if frekvens eq "a" and aggnivaa eq "knr" then &&
        lastvalue(bnpb.nr23_9.vr) &&
        else if frekvens eq "a" and aggnivaa eq "nr" then &&
        lastvalue(bnpb.nr2_.vr) &&
        else lastvalue(bnpb.nr23_9.vr.u))

if not (stap ge foerste_aar and slup le siste_aar)  -- som det finnes data for
type blank
type "Ulovlig beregningsperiode: " + &&
      string(year(stap)) + " - " + string(year(slup))

```

```

type "Databasen er oppdatert for perioden " + &&
  string(year(foerste_aar)) + " - " + string(year(siste_aar))
accept "Første beregningsår? ":stap
accept "Siste beregningsår? ":slup
end if

date stap to slup                                -- må ha lukket 'range' for å lage serier

type blank
type "Tidsrom for kjeding: " + @date
type blank

accept "Kjede hele databanken (j/n)? ":jn
if jn eq "j"

type "KJEDING AV HELE DATABANKEN STARTER NÅ, KL. " + now
type blank
if aggnivaa eq "knr"
  -/agg = "u"
  type blank
  type "KJEDER DISAGGREGERTE SERIER (KNR-sektor):"
  type blank
  $kjed_alle
  $sunntak
end if

if aggnivaa eq "nr"
  type "Kopierer summer som 'nrtot' (hjelpevariable)"
  $lag_nrtot
  type blank
  type "KJEDER PÅ AGGREGERT NIVÅ (nrpub):"
  type blank
else
  type blank
  type "KJEDER PÅ AGGREGERT NIVÅ (knrpub) ....."
  type blank
end if

if aggnivaa eq "knr"
  $lag_formler
end if

-/agg = "a"
$kjed_alle
$sunntak
$lag_blank

type blank
type "DATABANKEN ER FERDIG KJEDET, kl. " + now
type blank
type blank

if aggnivaa eq "knr"
  if frekvens eq "a"
    type blank
    type "Kopierer serier i løpende priser (LOKO, LON, ARBTP, FI, DRIB " + &&
      "og beregner NTAN (NTA + NSU)) fra knrgrlag til databasen med " + &&
      "kjedingsresultater for perioden " + @date + " pga. avstemming "+&&
      "mot årlig nr"
    type blank
    -/agg = "u"
    $kopier_fra_grlag
    -/agg = "a"
    $kopier_fra_grlag
  
```



```

    end if
  end if
  type blank
  type "Lager evt. alias 'nrtot' for alle totaler hvis kjeding ferdig "+&&
  "(pga. formler for omregning til valgfritt referanseår)"
  type blank
  accept "Lage alias (j/n)? ":jn
    if jn eq "j"
      $lagalias_nrtot
      type blank
    end if

  else if jn eq "n"
  type "BLOKKVIS KJEDING STARTER NÅ, KL. " + now
  type blank
  if aggnivaa eq "knr"
    accept "Uaggregert knr-sektor eller knr-pubagg (u/a)?" :agg
    if agg eq "a" or agg eq "A"
      $lag_formler
    end if
  else if aggnivaa eq "nr"
    accept "Uaggregert nr-sektor eller nr-pubagg (u/a)?" :agg
    if agg eq "u"
      type "OBS! Prosedyrekall er ikke lagt inn for alle variable enda!"
      type blank
    end if
  end if
  end if
  $kjed_en
end if

end block
END PROCEDURE

```

```
PROCEDURE $kjed_en
```

```

-/variabel = ""
accept "Variabel (tast 'h' for hjelp)?" : variabel

if (variabel eq "prod" or variabel eq "bil" or variabel eq "kons" or &&
variabel eq "inv" or variabel eq "eks" or variabel eq "imp" or &&
(variabel eq "dep" and frekvens eq "a") or &&
variabel eq "divagg" or variabel eq "bil" or variabel eq "lth" or &&
variabel eq "alle") and ((aggnivaa eq "knr") or &&
(aggnivaa eq "nr"))
  if agg eq "a" or agg eq "A"
    if crosstlist({BIF,BIL,BNPB,DEP,EKS,EKSB,IMP,KAPN,KAPB,KOH,KOI,KOO,NYI,&&
PIN,PINE,PINF,PROB,SBRK},{NRTOT},{VR,FP}) except &&
wildlist(work,"?nrtot?") ne {}
      type "Kopierer først summer som 'nrtot' (hjelpvariable)"
      type blank
      $lag_nrtot
    end if
  end if

-/prosedyre = "$kjed_" + string(variabel)
execute prosedyre
type blank
if variabel eq "alle" and aggnivaa eq "knr"
  $kopier_fra_grlag
  $sunntak
end if
if variabel eq "alle" and agg eq "a" or agg eq "A"
  accept "Lage alias til formler for valg av nytt referanseår (j/n)? ":jn
    if jn eq "j"

```

```

    $lagalias_nrtot
    type blank
  end if
end if
else
if variabel eq "h"
  $gi_hjelp
else
  type "Ugyldig valg: Variabel "+upper(string(variabel))+", "+ &&
    upper(string(aggnivaa))+ "-nivå, aggregeringsnivå " + &&
    upper(string(agg))
  end if
  type "Velg 'n' og variabel hvis du ønsker å fortsette"
  type blank
end if

```

```
$ferdig
```

```
END PROCEDURE -- $skjed_en
```

```
-----
PROCEDURE $skjed_alle
-----
```

```

LOOP for x in &&
  (if frekvens eq "a" then alle_serier else alle_serier except {dep})

  -/alle_variable = lower(name(x))
  -/prosedyre = "$skjed_" + alle_variable + ""
  execute prosedyre
END LOOP

```

```
END PROCEDURE -- $skjed_alle
```

```
-----
PROCEDURE $lag_blank
-----
```

```

-/blank_var_VRogVL = {PROB, BNPB, PIN, PINE, PINF, KOH, KOO, KOI, EKS, EKSB, &&
  IMP, BIF, NYI, SBRK, BIL, DEP, KAPN, KAPB}
-/blank_var_bareVR = {DRIB, FI, LOKO, LON, ARBTP, NTA, NSU, NTAN}

```

```
if @store eq "UTDAT"
```

```

  LOOP for x in blank_var_VRogVL union blank_var_bareVR
    utdat'x&blank&vl = series(NA)
    utdat'x&blank&vr = series(NA)
  END LOOP

```

```
else if @store eq "VELG_REFAAR"
```

```

  LOOP for x in blank_var_VRogVL
    velg_refaar'x&blank&vlr = series(NA)
  END LOOP

```

```
end if
```

```
END PROCEDURE
```

```
PROCEDURE $gi_hjelp
```

```

type "Kjeding og oppdatering av formler for valgfritt"
type "referanseår kan utføres enten for:"
type " Hele databanken under ett, agg 'a' OG 'u' (velg 'j')"
type " Alle variable, agg 'a' ELLER 'u' , (velg 'n', 'alle')
type " Blokkvis (velg 'n', blokk)"
type blank
type "Mulighetene er:"
type !{alle} union alle_serier
type blank
type "Aggregeringsnivå:"
type " - Uaggregert KNR (knr,u)"
type " - KNR-pubagg (knr,a)"
type " - Uaggregert NR (nr,u)- foreløpig ikke lagt inn alt"
type " - NR-pubagg (nr,a)"
type blank

```

```
END PROCEDURE -- $gi_hjelp
```

```
PROCEDURE $ferdig
```

```
-- hjelpeprosedyre, kalles av $kjed og $velg_refaar
```

```

accept "Ferdig (j/n)? ":jn
if jn eq "n"
  if @store eq "UTDAT"
    $kjed_en
  else if @store eq "VELG_REFAAR"
    $velg_refaar_en
  end if
else if jn eq "j"
  if @store eq "UTDAT"
    type "BLOKKVIS KJEDING FERDIG NÅ, KL. " + now
    return
  else if @store eq "VELG_REFAAR"
    type "BLOKKVIS OPPDATERING AV FORMLER FERDIG NÅ, KL. " + now
    type "NYTT REFERANSEÅR ER: " + string(year(referanse_aar))
    type "Om du ønsker å skrive ut data med et annet referanseår, gi "+&&
      "(f.eks) følgende kommando: -/referanse_aar = 92 "
    return
  end if
else
  type "Svar 'j' eller 'n'!"
  type " "
  type " "
  $ferdig
end if

```

```
END PROCEDURE -- $ferdig
```

```
PROCEDURE $unntak
```

```

type blank
type "Beregner evt. unntak fra generell formel til slutt"
type "$unntak_eks forutsetter at PROB og SBRK er kjedet"
type "$unntak_inv forutsetter at EKS, NYI og SBRK er kjedet"
$unntak_inv
$unntak_eks
type "UNNTAK FERDIG BEREGNET, kl. " + now

```

```
END PROCEDURE -- $unntak
```

```
PROCEDURE $kjed_prod
```

```
-- til kjeding av vareinnsats brukes nrlnrps29 istedenfor nrlnrps på
-- detaljert knr-nivå pga. avstemmingen, da knr-databanken inneholder
-- sektor nr2958
```

```
type "PRODUKSJONSSEKTORER, "+&&
upper(string(aggntvaa))"-nivå, aggregeringsnivå: " + upper(string(agg))
```

```
loop for i in {vr,fp}
  -/z=name(i)
```

```
-- Bruttoprodukt og produktinnsats uten korreksjonsposter --
```

```
execute "formula work'prob.nr23U."+z+"= prob.nr23_."+z + &&
"- prob.nr23654."+z+" - prob.nr23659."+z+" - prob.nr23669."+z
```

```
execute "formula work'pin.nr23U."+z+"= pin.nr23_."+z + &&
"- pin.nr23654."+z+" - pin.nr23659."+z+" - pin.nr23669."+z
```

```
execute "formula work'bnpb.nr23U."+z+"= bnpb.nr23_."+z + &&
"- bnpb.nr23654."+z+" - bnpb.nr23659."+z+" - bnpb.nr23669."+z
```

```
-- Ikke-markedsrettet virksomhet--
```

```
execute "formula work'bnpb.nrimark."+z+"=overlay(("&&
"bnpb.nr22_."+z+" + bnpb.nr24_."+z + &&
"+ bnpb.nr25_."+z+" - bnpb.nr25410."+z+"- "&&
"bnpb.nr25900."+z+" + bnpb.nr26_."+z+"), "&&
"prob.nrimark."+z+" - "&&
"pin.nrimark."+z+")"
```

```
-- Markedsrettet virksomhet--
```

```
execute "formula work'prob.nrimark."+z+"= " + &&
"overlay((prob.nr23U."+z + &&
"+ prob.nr25410."+z+" + prob.nr25900."+z+"), "&&
"prob.nr23_6."+z+" - prob.nrimark."+z+" - "&&
"prob.nr2368."+z+")"
```

```
execute "formula work'pin.nrimark."+z+"= " + &&
"overlay((pin.nr23U."+z + &&
"+ pin.nr25410."+z+" + pin.nr25900."+z+"), "&&
"convert(pin.nr23_6."+z+" - pin.nrimark."+z+" - "&&
"pin.nr2368."+z+")"
```

```
execute "formula work'bnpb.nrimark."+z+"= " + &&
"overlay((bnpb.nr23U."+z + &&
"+ bnpb.nr25410."+z+" + bnpb.nr25900."+z+"), "&&
"convert(bnpb.nr23_6."+z+" - bnpb.nrimark."+z+" - "&&
"bnpb.nr2368."+z+")"
```

```
-- Hjelpe-formler
```

```
execute "formula work'prob.nr23654."+z+"= bnpb.nr23654."+z + &&
"+ pin.nr23654."+z
```

```
execute "formula work'prob.nr23659."+z+"= bnpb.nr23659."+z + &&
"+ pin.nr23659."+z
```

```
execute "formula work'prob.nr23669."+z+"= bnpb.nr23669."+z + &&
"+ pin.nr23669."+z
```

```
end loop
```

```

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

-- Til trend-bergninger i KNR
formula work'prob.nr2385b.vr=prob.nr2385.vr-prob.nr23851.vr
formula work'prob.nr2390c.vr=prob.nr2390.vr- &&
    (prob.nr23922.nr922001.vr + prob.nr23927.vr)
formula work'prob.nr2690b.vr=prob.nr2690.vr-prob.nr26921.vr

formula work'prob.nr2385b.fp=prob.nr2385.fp-prob.nr23851.fp
formula work'prob.nr2390c.fp=prob.nr2390.fp- &&
    (prob.nr23922.nr922001.fp + prob.nr23927.fp)
formula work'prob.nr2690b.fp=prob.nr2690.fp-prob.nr26921.fp

$skjed_sektor nrlknrps29 union {nr23603,nr2385b,nr2390c,nr2690b}, "PROB", "U"
$skjed_sektor nrlknrva, "PROB", "U"
$skjed_sektor nrlknrps, "PINE", "U"
$skjed_sektor nrlknrps, "PINF", "U"
$skjed_sektor nrlknrps29, "PIN", "U"
$skjed_sektor nrlknrps29, "BNPB", "U"

else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

-/aggps=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggps.txt | grep -i 'nr'",case))
-/aggps29=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggps29.txt | grep -i 'nr'",case))
-/aggpin=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggpin.txt | grep -i 'nr'",case))
-/aggdivps = nl(system("grep -i 'bnpb.nr' "+&&
"$NASJREGN/knr/tabell/tab_makro.txt | cut -c 8-25",case)) except {nr29korr}

$skjed_sektor aggps29 union {nr29prod,nr23_6,nrprim}, "BNPB", "A"
$skjed_sektor aggdivps, "BNPB", "A"
$skjed_sektor aggps union aggpin, "PIN", "A"
$skjed_sektor {nr2_80,nr2_85,nrimark,nrmark}, "PIN", "A"
$skjed_sektor aggdivps union {nrprim} except {nr23_9,nr23_9fn}, "PIN", "A"
$skjed_sektor aggps union {nr2_85,nrimark,nrmark}, "PROB", "A"
$skjed_sektor {nr051_2,nrprim,nrprim,nrjord,nr23vare}, "PROB", "A"

-- Produksjon, gebyrer, stat og kommuneforvaltningen
-- Til Pål Drevland/Teknisk Beregningsutvalg

-/agg24xx = {nr757,nr807,nr857,nr907}
-/agg25xx = {nr419,nr809,nr859,nr909}

work'prob.nr24geb.vr = lsum(crosslist({prob},agg24xx,{vr}))
work'prob.nr24geb.fp = lsum(crosslist({prob},agg24xx,{fp}))
work'prob.nr25geb.vr = lsum(crosslist({prob},agg25xx,{vr}))
work'prob.nr25geb.fp = lsum(crosslist({prob},agg25xx,{fp}))

$skjed_sektor {nr24geb,nr25geb}, "PROB", "A"

else if aggnivaa eq "nr" and (agg eq "u" or agg eq "U")

-/nrn = system("grep -i '^2' $NASJREGN/tidsserier/meta/mottaker.txt", case)
-/nrrips = nl("NR" + substring(nrn,1, location(nrn, " ") -1))

$skjed_sektor nrrips, "PROB", "U"          -- bruttoproduksjon
$skjed_sektor nrrips, "BNPB", "U"        -- bruttoprodukt
$skjed_sektor nrrips, "PIN", "U"         -- produktinnsats

else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")

```

```

-- Spesial-aggregater til Ingunn (rapp.)
work'prob.skog.vr =prob.nr2302.vr+prob.nr2320.vr+prob.nr2321.vr
work'prob.skog.fp =prob.nr2302.fp+prob.nr2320.fp+prob.nr2321.fp
work'prob.kjemisk.vr =prob.nr2324.vr+prob.nr2325.vr
work'prob.kjemisk.fp =prob.nr2324.fp+prob.nr2325.fp
work'prob.IT.vr=prob.nr23300.vr+prob.nr23720.vr
work'prob.IT.fp=prob.nr23300.fp+prob.nr23720.fp
work'prob.telekomm.vr=prob.nr23321.vr+prob.nr23642.vr
work'prob.telekomm.fp=prob.nr23321.fp+prob.nr23642.fp

work'pin.skog.vr =pin.nr2302.vr+pin.nr2320.vr+pin.nr2321.vr
work'pin.skog.fp =pin.nr2302.fp+pin.nr2320.fp+pin.nr2321.fp
work'pin.kjemisk.vr =pin.nr2324.vr+pin.nr2325.vr
work'pin.kjemisk.fp =pin.nr2324.fp+pin.nr2325.fp
work'pin.IT.vr=pin.nr23300.vr+pin.nr23720.vr
work'pin.IT.fp=pin.nr23300.fp+pin.nr23720.fp
work'pin.telekomm.vr=pin.nr23321.vr+pin.nr23642.vr
work'pin.telekomm.fp=pin.nr23321.fp+pin.nr23642.fp

work'bnpb.skog.vr =bnpb.nr2302.vr+bnpb.nr2320.vr+bnpb.nr2321.vr
work'bnpb.skog.fp =bnpb.nr2302.fp+bnpb.nr2320.fp+bnpb.nr2321.fp
work'bnpb.kjemisk.vr =bnpb.nr2324.vr+bnpb.nr2325.vr
work'bnpb.kjemisk.fp =bnpb.nr2324.fp+bnpb.nr2325.fp
work'bnpb.IT.vr=bnpb.nr23300.vr+bnpb.nr23720.vr
work'bnpb.IT.fp=bnpb.nr23300.fp+bnpb.nr23720.fp
work'bnpb.telekomm.vr=bnpb.nr23321.vr+bnpb.nr23642.vr
work'bnpb.telekomm.fp=bnpb.nr23321.fp+bnpb.nr23642.fp

-/aggingunns = {skog,kjemisk,IT,telekomm}
-/aggpublishprob = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/Gmlforsp/tab_pubhov_prob.txtlgrep -i 'nr'",case))
-/aggpublishpin = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubnaer_prob.txt | grep -i 'nr'",case))
-/aggpublishpin = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/Gmlforsp/tab_pubhov_pin.txtlgrep -i 'nr'",case))
-/aggpublishpin = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubnaer_pin.txt | grep -i 'nr'",case))
-/aggpublishbnpb = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/Gmlforsp/tab_pubhov_bnpb.txtlgrep -i 'nr'",case))
-/aggpublishbnpb = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubnaer.txt | grep -i 'nr'",case))

-- Næring
$kjed_sektor aggpublishprob, "PROB", "A"      -- bruttoproduksjon
$kjed_sektor aggpublishpin, "PIN", "A"      -- vareinnsats i alt
$kjed_sektor aggpublishbnpb, "BNPB", "A"    -- bruttoprodukt
$kjed_sektor aggingunns, "PROB", "A"
$kjed_sektor aggingunns, "PIN", "A"
$kjed_sektor aggingunns, "BNPB", "A"

-- Hovednæring
$kjed_sektor aggpublishprob, "PROB", "A"
$kjed_sektor aggpublishpin, "PIN", "A"
$kjed_sektor aggpublishbnpb, "BNPB", "A"

end if

type "Produksjonssektorene er ferdig kjedet, kl. " + now
type blank

END PROCEDURE -- $kjed_prod

```

PROCEDURE \$kjed_kons

```

type "KONSUM, "+&&
upper(string(aggnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

-- Til trend-beregninger i KNR
work'koh.nr61941.nto.vr = koh.nr61941.vr - koh.nr61941.gebyr.vr
work'koh.nr61942.nto.vr = koh.nr61942.vr - koh.nr61942.gebyr.vr
work'koh.nr61961.nto.vr = koh.nr61961.vr - koh.nr61961.gebyr.vr
work'koh.nr6155.atransp.vr = &&
    koh.nr61531.nr602210.vr + koh.nr61531.nr611011.vr + &&
    koh.nr61531.nr611028.vr + koh.nr61532.nr611014.vr + &&
    koh.nr61532.nr660319.vr + koh.nr61533.nr602410.vr
work'koh.nr61941.nto.fp = koh.nr61941.fp - koh.nr61941.gebyr.fp
work'koh.nr61942.nto.fp = koh.nr61942.fp - koh.nr61942.gebyr.fp
work'koh.nr61961.nto.fp = koh.nr61961.fp - koh.nr61961.gebyr.fp
work'koh.nr6155.atransp.fp = &&
    koh.nr61531.nr602210.fp + koh.nr61531.nr611011.fp + &&
    koh.nr61531.nr611028.fp + koh.nr61532.nr611014.fp + &&
    koh.nr61532.nr660319.fp + koh.nr61533.nr602410.fp

-/konstrend = {nr61941.nto,nr61942.nto,nr61961.nto,nr6155.atransp,nr61525,&&
    nr61619,nr61624,nr61911}

$kjed_sektor nrlknrep union konstrend, "KOH", "U"
$kjed_sektor_art {nr6151},{nr0943}, "KOH", "A"
$kjed_sektor nrlknrci union {nr66tjen}, "KOI", "U"
$kjed_sektor nrlknrg, "KOO", "U"

else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

-/aggcp=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggcp.txt | grep -i 'nr'",case))
-/aggkons=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggkons.txt | grep -i 'nr'",case))

$kjed_sektor aggcp, "KOH", "A"
$kjed_agg aggkons, "A"

else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")

-/aggpubkoh = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubkoh.txt | grep -i 'nr'",case))
-/aggpubkoi = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubkoide.txt | grep -i 'nr'",case))
-/aggpubkoo = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubkoo.txt | grep -i 'nr'",case))
-/aggpubkos=nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubkos.txt | grep -i 'nr'",case))
-/aggpubkok=nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubkok.txt | grep -i 'nr'",case))
-/aggkons=nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_aggkons.txt | grep -i 'nr'",case))

$kjed_sektor aggpubkoh, "KOH", "A"
$kjed_sektor aggpubkoo, "KOO", "A"
$kjed_sektor aggpubkoi, "KOI", "A"
$kjed_sektor aggpubkos, "KOO", "A"
$kjed_sektor aggpubkok, "KOO", "A"
$kjed_agg aggkons, "A"

```

end if

type "Konsum ferdig kjedet, kl. " + now
type blank

END PROCEDURE -- \$kjed_kons

PROCEDURE \$kjed_eks

type "EKSPORT, "+&&
upper(string(agnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

-/nrlnrjr09= {nr0915,nr0937,nr0938,nr0939,nr0941,nr0942,nr0943, &&
nr0955,nr0971}

\$kjed_sektor nrlnrva union nrlnrjr09, "EKS", "U"
\$kjed_sektor nrlnrva union nrlnrjr09, "EKSB", "U"

else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

-/aggeks=nl(system("cut -c 3-20 "+&&
"\$NASJREGN/knr/tabell/tab_aggeks.txt | grep -i 'nr"',case))
-/aggeksb = {nrtot,nr051_2,nrprimp,nrjord}

\$kjed_sektor aggeks, "EKS", "A"
\$kjed_sektor aggeksb, "EKSB", "A"

else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")

-- Spesial-aggregater til Ingunn (rapp.)
work'eks.skog.vr =eks.nr020.vr+eks.nr200.vr+eks.nr210.vr
work'eks.skog.fp =eks.nr020.fp+eks.nr200.fp+eks.nr210.fp
work'eks.kjemisk.vr =eks.nr247.vr+eks.nr251.vr
work'eks.kjemisk.fp =eks.nr247.fp+eks.nr251.fp
work'eks.fisk.vr=eks.nr051.vr+eks.nr052.vr
work'eks.fisk.fp=eks.nr051.fp+eks.nr052.fp

series <store work; case 1 to *> alle_eks : string by case

-/i = 1
LOOP for x in &&
 wildlist(nr02,"eks?vr")
 set alle_eks[i] = replace(replace(string(name(x)), ".VR", "",last),&&
 "EKS.", "",first)
 set i = i+1
END LOOP

-/aggeks = nl(alle_eks)
-/aggeksingunns = {skog,kjemisk,fisk}

\$kjed_sektor aggeks, "EKS", "A"
\$kjed_sektor aggeksingunns, "EKS", "A"

end if

type "Eksport er ferdig kjedet, kl. " + now
type blank

END PROCEDURE -- \$kjed_eks

```
PROCEDURE $kjed_imp
```

```

type "IMPORT, "+&&
upper(string(agnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

    $kjed_sektor nrlknrva, "IMP", "U"

else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

    -/aggimp=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggimp.txt | grep -i 'nr"',case)) union &&
    {nr051_2,nrprimp,nrjord,nrdimpa}

    $kjed_sektor aggimp, "IMP", "A"

else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")

    series <store work; case 1 to *> alle_imp : string by case

    -/i = 1
    LOOP for x in &&
        wildlist(nr02,"imp?vr")
        set alle_imp[i] = replace(replace(string(name(x)), ".VR", "", last), &&
            "IMP.", "", first)
        set i = i+1
    END LOOP

    -/aggimp = nl(alle_imp)

    $kjed_sektor aggimp, "IMP", "A"

end if

type "Import er ferdig kjedet, kl. " + now
type blank

END PROCEDURE -- $kjed_imp

```

```
PROCEDURE $kjed_inv
```

```

-- bruttoinvestering i fast kapital = nyinv. + salg av brukt realkapital
-- kjeder bruttoinvesteringer, nyinvesteringer og nto. salg av brukt real-
-- kapital hver for seg på samme måte i års- og kvartalsregnskapet (BIF blir
-- ikke lik sum NYI og SBRK, men avviket godtas fordi SBRK ikke publiseres)

type "INVESTERINGSSEKTOR OG INVESTERINGSEKTOR*ART, "
type upper(string(agnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

    $kjed_sektor nrlknrjs, "SBRK", "U"
    $kjed_sektor nrlknrjr, "SBRK", "U"
    $kjed_sektor_art nrlknrjs, nrlknrjr, "SBRK", "U"

type "SBRK ferdig kjedet, kl. " + now

```

```
$kjed_sektor nrlknrjs, "NYI", "U"
$kjed_sektor nrlknrjr, "NYI", "U"
$kjed_sektor_art nrlknrjs, nrlknrjr, "NYI", "U"
```

```
type "NYI er ferdig kjedet, kl. " + now
```

```
$kjed_sektor nrlknrjs, "BIF", "U"
$kjed_sektor nrlknrjr, "BIF", "U"
$kjed_sektor_art nrlknrjs, nrlknrjr, "BIF", "U"
```

```
type "BIF er ferdig kjedet, kl. " + now
```

```
else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")
```

```
-/aggjbm = {nr08b,nr08m}
-/aggjs=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggjs.txt | grep -i 'nr' ",case)) union {nrprim}
-/aggjsjr=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggjsjr.txt | grep -i 'nr' ",case)) union {nrprim}
-/aggdivjs = nl(system("grep -i 'bif.nr' "+&&
"$NASJREGN/knr/tabell/tab_makro.txt | cut -c 7-25",case)) union &&
{nr83oljsj,nr83_,nr86_}
```

```
$kjed_sektor aggjsjr except {nr8368}, "SBRK", "A"
$kjed_sektor_art nrlknrjs, aggjbm, "SBRK", "A"
$kjed_sektor_art aggjs except {nr8368}, aggjbm, "SBRK", "A"
$kjed_sektor aggdivjs except {nr8_} union aggjbm, "SBRK", "A"
$kjed_sektor_art aggdivjs except {nr8_}, aggjbm, "SBRK", "A"
```

```
type "SBRK er ferdig kjedet, kl. " + now
```

```
$kjed_sektor aggjsjr except {nr8368}, "NYI", "A"
$kjed_sektor_art nrlknrjs, aggjbm, "NYI", "A"
$kjed_sektor_art aggjs except {nr8368}, aggjbm, "NYI", "A"
$kjed_sektor aggdivjs except {nr8_} union aggjbm, "NYI", "A"
$kjed_sektor_art aggdivjs except {nr8_}, aggjbm, "NYI", "A"
```

```
type "NYI er ferdig kjedet, kl. " + now
```

```
$kjed_sektor aggjsjr except {nr8368}, "BIF", "A"
$kjed_sektor_art nrlknrjs, aggjbm, "BIF", "A"
$kjed_sektor_art aggjs except {nr8368}, aggjbm, "BIF", "A"
$kjed_sektor aggdivjs union aggjbm, "BIF", "A"
$kjed_sektor_art aggdivjs except {nr8_}, aggjbm, "BIF", "A"
```

```
type "BIF er ferdig kjedet, kl. " + now
```

```
else if aggnivaa eq "nr" and (agg eq "u" or agg eq "U")
```

```
-/nrm = system("grep -i '^8' $NASJREGN/tidsserier/meta/mottaker.txt", case)
-/nrljs = nl( "NR" + substring(nrm,1, location(nrm,"")-1) )
-/nrp008 = system("egrep -i '^008' "+&&
"$NASJREGN/tidsserier/meta/produkt.txt", case)
-/nrljr008 = nl( "NR" + substring(nrp008,1, location(nrp008,"")-1) )
-/nrp009 = system("egrep -i '^009' "+&&
"$NASJREGN/tidsserier/meta/produkt.txt", case)
-/nrljr009 = nl( "NR" + substring(nrp009,1, location(nrp009,"")-1) )
```

```
$kjed_sektor nrljs, "BIF", "U"
--$kjed_sektor_art nrljs, nrljr008, "BIF", "U"
--$kjed_sektor_art nrljs, nrljr009, "BIF", "U"
```

```
else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")
```

```

-- Spesial-aggregater til Ingunn (rapp.)
work'bif.skog.vr =bif.nr8302.vr+bif.nr8320.vr+bif.nr8321.vr
work'bif.skog.fp =bif.nr8302.fp+bif.nr8320.fp+bif.nr8321.fp
work'bif.kjemisk.vr =bif.nr8324.vr+bif.nr8325.vr
work'bif.kjemisk.fp =bif.nr8324.fp+bif.nr8325.fp
work'bif.IT.vr=bif.nr83300.vr+bif.nr83720.vr
work'bif.IT.fp=bif.nr83300.fp+bif.nr83720.fp
work'bif.telekomm.vr=bif.nr83321.vr+bif.nr83642.vr
work'bif.telekomm.fp=bif.nr83321.fp+bif.nr83642.fp

-/aggingunns = {skog,kjemisk,IT,telekomm}
-/agppubinv = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubinv.txt | grep -i 'nr' ",case))
-/agppubinvart = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubinvart.txt | grep -i 'nr' ",case))

$skjed_sektor agppubinv, "BIF", "A"
$skjed_sektor agppubinvart, "BIF", "A"
$skjed_sektor aggingunns, "BIF", "A"

type "BIF er ferdig kjedet, kl. " + now

end if

type "Alle investeringssektorer/arter er ferdig kjedet, kl. " + now
type blank

END PROCEDURE -- $skjed_inv

```

```
PROCEDURE $skjed_dep
```

```
type "KAPITALSLIT OG KAPITALBEHOLDNING ETTER SEKTOR, "+&&
upper(string(agnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))
```

```
if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")
```

```

$skjed_sektor nrlknrps, "DEP", "U"
$skjed_sektor_art nrlknrps, nrlknrjr, "DEP", "U"
$skjed_sektor nrlknrjr, "DEP", "U"

-- Netto
$skjed_sektor nrlknrps, "KAPN", "U"
$skjed_sektor_art nrlknrps, nrlknrjr, "KAPN", "U"
$skjed_sektor nrlknrjr, "KAPN", "U"

-- Brutto - mangler på knr-nivå
--$skjed_sektor nrlknrps, "KAPB", "U"
--$skjed_sektor_art nrlknrps, nrlknrjr, "KAPB", "U"
--$skjed_sektor nrlknrjr, "KAPB", "U"

```

```
else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")
```

```

-/aggjbm = {nr08b,nr08m}
-/aggps=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggps.txt | grep -i 'nr'",case))

$skjed_sektor aggps except {nrimark,nrmark}, "DEP", "A"
--$skjed_sektor_art nrlknrps, aggjbm, "DEP", "A" -- mangler formler
--$skjed_sektor_art aggps, aggjbm, "DEP", "A" -- mangler formler

```

```

-- Netto
$skjed_sektor aggps, "KAPN", "A"
--$skjed_sektor_art nrlnrps, aggjbm, "KAPN", "A" -- mangler formler
--$skjed_sektor_art aggps, aggjbm, "KAPN", "A" -- mangler formler

-- Brutto - mangler på knr-nivå
--$skjed_sektor aggps, "KAPB", "A"
--$skjed_sektor_art nrlnrps, aggjbm, "KAPB", "A" -- mangler formler
--$skjed_sektor_art aggps, aggjbm, "KAPB", "A" -- mangler formler
--$skjed_sektor aggps, "KAPB", "A"

else if aggnivaa eq "nr" and (agg eq "u" or agg eq "U")

-/nrs = system("grep -i '^2' $NASJREGN/tidsserier/meta/mottaker.txt", case)
-/nrtps = nl( "NR" + substrng(nrs,1, location(nrs," ")-1) )

$skjed_sektor nrtps, "DEP", "U"
$skjed_sektor nrtps, "KAPN", "U"
$skjed_sektor nrtps, "KAPB", "U"

else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")

-/agppubn = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubnaer.txt | grep -i 'nr'",case))

$skjed_sektor agppubn, "DEP", "A"
$skjed_sektor agppubn, "KAPN", "A"
$skjed_sektor agppubn, "KAPB", "A"

end if

type "Kapitalslit og kapitalbeholdning er ferdig kjedet, kl. " + now
type blank

END PROCEDURE -- $skjed_dep

```

```
PROCEDURE $skjed_lth
```

```

type "LEVERANSE TIL HJEMMEMARKEDET, "+&&
upper(string(aggnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

if frekvens eq "a"
type "Leveranse til hjemmemarkedet beregnes ikke i årsregnskapet"
end if

if aggnivaa eq "knr"

-- LTH disagg. må beregnes uansett da data ikke lages i årlig NR
if wilddlist(work,"lth?vr") except {lth.nrtot.vr} eq {}
type "Beregner disagg. LTH som diff. PROB-EKS pga. KNR"
$skjed_sektor nrlnrva, "LTH", "U"
end if

if agg eq "a" or agg eq "A"
type "Bruker beregnet LTH (diff. PROB-EKS) og lager aggregater "+&&
"til KNR. FP/VR lagres i work, VL/VR lagres i "+ @store

-- Brukes i KNR/div. aggregeringsformler (lth?, iesp? og ianv?)
work'lth.nr051_2.vr = lsum(crosslist({lth},{nr051,nr052},{vr}))
work'lth.nrjord.vr = lsum(crosslist({lth},{nr011,nr012,nr019},{vr}))
work'lth.nrprimp.vr = lsum(crosslist({lth}, &&
{nr011,nr012,nr019,nr051,nr052,nr020},{vr}))

```

```

work'lth.nrtot.vr = lsum(crosslist({lth}, nrknrva except &&
    {nr756,nr806,nr854,nr906,nr808,nr858,nr908},{vr}))
work'lth.nr051_2.fp = lsum(crosslist({lth},{nr051,nr052},{fp}))
work'lth.nrjord.fp = lsum(crosslist({lth},{nr011,nr012,nr019},{fp}))
work'lth.nrprimp.fp = lsum(crosslist({lth}, &&
    {nr011,nr012,nr019,nr051,nr052,nr020},{fp}))
work'lth.nrtot.fp = lsum(crosslist({lth}, nrknrva except &&
    {nr756,nr806,nr854,nr906,nr808,nr858,nr908},{fp}))

```

-- Formler som benytter LTH

```

work'ianv.nr020.vr = lth.nr020.vr + imp.nr020.vr
work'ianv.nr051_2.vr = lth.nr051_2.vr + imp.nr051_2.vr
work'ianv.nrjord.vr = lth.nrjord.vr + imp.nrjord.vr
work'ianv.nrprimp.vr = lth.nrprimp.vr + imp.nrprimp.vr
work'ianv.nr020.fp = lth.nr020.fp + imp.nr020.fp
work'ianv.nr051_2.fp = lth.nr051_2.fp + imp.nr051_2.fp
work'ianv.nrjord.fp = lth.nrjord.fp + imp.nrjord.fp
work'ianv.nrprimp.fp = lth.nrprimp.fp + imp.nrprimp.fp
work'iesp.nr020.vr = ianv.nr020.vr - bil.nr020.vr
work'iesp.nr051_2.vr = ianv.nr051_2.vr - bil.nr051_2.vr
work'iesp.nrjord.vr = ianv.nrjord.vr - bil.nrjord.vr
work'iesp.nrprimp.vr = ianv.nrprimp.vr - bil.nrprimp.vr
work'iesp.nr020.fp = ianv.nr020.fp - bil.nr020.fp
work'iesp.nr051_2.fp = ianv.nr051_2.fp - bil.nr051_2.fp
work'iesp.nrjord.fp = ianv.nrjord.fp - bil.nrjord.fp
work'iesp.nrprimp.fp = ianv.nrprimp.fp - bil.nrprimp.fp

```

```

-/aggllth={nrtot,nr051_2,nrprimp,nrjord}
-/aggianv_iesp = &&
    crosslist({ianv,iesp},{nr020,nr051_2,nrjord,nrprimp})

```

```

$skjed_sektor aggllth, "LTH", "A"
$skjed_agg_ianv_iesp, "A"

```

end if

```

type "Leveranse til hjemmemarkedet ferdig kjedet, kl. " + now
type blank

```

```

else
    type "Beregningsopplegg er ikke implementert"
end if

```

END PROCEDURE -- \$skjed_lth

PROCEDURE \$skjed_bil

```

-- Bruker LTH, leveranse til hjemmemarkedet, for samme knr-produkt til
-- kjeding av lagertallene. I årsregnskapet beregnes LTH som differansen
-- produksjon - eksport (lagres midlertidig), og denne kjedeindeksen benyttes
-- i stedet for lager-indeksen
-- I knr beregnes LTH i modellen. Pga. varebalanseringen i knr, kan man ikke
-- ha lager av tjenester -> ufordelt lagerendring NB! Årssummen av serien skal
-- helst være lik 0 (ufv etter knr-sektor er lik 0 fram til basisåret, og opp-
-- rettes som resultat av modell-kjøringene,- serien kjedes derfor ikke.
-- Summen, ufv.nrtot, er lik bil.nr8L (u999 = ds990), og kjedes med indeksen
-- for LTH totalt
-- Alle aggregater for lager (knr og nr), kjedes med totalindeksen for LTH

```

```

type "LAGERENDRING, UFORDELT LAGER, "+&&
upper(string(aggnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))
if aggnivaa eq "nr"
    type "Prosedyrekall er ikke lagt inn enda!"

```

```
type blank
```

```
else
```

```
if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")
```

```
    $skjed_sektor nrknrva, "BIL", "U"
```

```
else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")
```

```
    $skjed_sektor {nr8lu_nr8l_nr051_2_nrprimp,nrjord}, "BIL", "A"
    $skjed_sektor {nrtot}, "UFV", "U"
```

```
end if
```

```
type "Lagerendring og ufordelt lager ferdig kjedet, kl. " + now
type blank
```

```
end if
```

```
END PROCEDURE -- $skjed_bil
```

```
PROCEDURE $skjed_divagg
```

```
type "DIV. AGGREGATER, "+&&
```

```
upper(string(aggnivaa))+ "-nivå, aggregeringsnivå: " + upper(string(agg))
```

```
if agg eq "u" or agg eq "U"
```

```
    type "Finnes bare på agg-nivå 'a'"
```

```
    type blank
```

```
else
```

```
if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")
```

```
    -/aggmakro = nl(system("cut -c 3-20 "+&&
    "$NASJREGN/knr/tabell/tab_makro.txt |grep -i 'nr' ",case))
    -/agganvendelse = nl(system("cut -c 3-20 "+&&
    "$NASJREGN/knr/tabell/tab_anvendelse.txt | grep -i 'nr' ",case))
    -/aggfindep=nl(system("cut -c 3-15 "+&&
    "$NASJREGN/tidsserier/tabell/tab_kjedextra.txt", case))
```

```
    $skjed_agg aggmakro, "A"
```

```
    $skjed_agg agganvendelse, "A"
```

```
    $skjed_agg aggfindep, "A"
```

```
    $skjed_sektor {nr8lu_nr8l_}, "BIL", "A" -- bruker LTH
```

```
type "Makrøsk. hovedstørrelser er ferdig kjedet, kl. " + now
```

```
type "BNP etter anvendelse er ferdig kjedet, kl. " + now
```

```
type blank
```

```
else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")
```

```
    -/aggpubtilg = nl(system("cut -c 3-20 "+&&
    "$NASJREGN/tidsserier/tabell/tab_pubtilg.txt | grep -i 'nr' ",case)) &&
    union {tot.tilgialt,tot.anvialt}
```

```
    $skjed_agg aggpubtilg, "A"
```

```
type "Tilgang og anvendelse er ferdig kjedet, kl. " + now
```

```
type blank
```

```
end if
```

```
end if
```

```
END PROCEDURE -- $skjed_divagg
```

Vedlegg 2: Kjedeformelen \$kjed_sektor

```
-- Programmert av Pia E. Tønjum, 95/96
-- Sist endret; PET,28/10-96
-- Sist endret; PET,17/2-98 (lagt inn 95=100)
-- Kjeder etter sektor eller produkt - loop gjennom 1 liste
-- indeksene som brukes til kjedingen beregnes først - bruker års(pris)indeks
-- som kjedeindeks, og nivå tallet løpende i priser i beregningsåret
-- (dersom kjedeindeksen = 0 brukes isteden totalindeksen for samme variabel)
-- For kjeding av lager brukes prisindeksen for leveranse til hjemmemarkedet
-- for samme produkt (til lager totalt benyttes totalindeksen for LTH)
```

```
PROCEDURE $kjed_sektor -- generell kjedeformel, kalles av $kjed
```

```
argument liste, &&      -- listekode, f.eks. nrlknrps, nrlknrva
    var, &&              -- variabelnavn, f.eks. BNPB, LTH
    agg                 -- aggnivå: uagg (U) eller agg (A)
```

```
block
store utdat
over on
glue dot
date stap to slup
ignore addition on
ignore function on
```

```
----- Kvartalstall -----
```

```
if frekvens eq "q"
```

```
if agg eq "U"           -- til totalindeksen (summerer opp ukjedede tall)
if not exists(work'id(var)&nrtot&vr&u)
    work'id(var)&nrtot&vr&u = lsum(selectnames(crosslist({id(var)},liste,&&
        {vr&u}),exists(@name)))
    work'id(var)&nrtot&fp&u = lsum(selectnames(crosslist({id(var)},liste,&&
        {fp&u}),exists(@name)))
end if
end if
```

```
local index=0
LOOP for x in (if var eq "LTH" then liste union {nrtot} else liste)
    set index=index+1
```

```
if agg eq "U" or agg eq "u"
local new l&x = if not exists(id(var)&x&vr&u) then series(0) &&
    else overlay(id(var)&x&vr&u,series(0))
```

```
local new f&x = if not exists(id(var)&x&fp&u) then series(0) &&
    else overlay(id(var)&x&fp&u,series(0))
```

```
else if agg eq "A" or agg eq "a"
if not exists(id(var)&x&vr&u)
    type "Aggregat mangler for "+name(id(var)&x)
    write "Aggregat mangler for "+name(id(var)&x) to feilmelding
    next
else
    local new l&x = overlay(id(var)&x&vr&u,series(0))
    local new f&x = overlay(id(var)&x&fp&u,series(0))
end if
end if
```

```

local new p&x = if abs(convert(l&x,a,*,sum)) ge 1 then &&
                convert(l&x,a,*,sum)/ &&
                convert(f&x,a,*,sum) else &&
                (if var eq "BIL" and agg eq "U" or agg eq "u" then &&
                 (if abs(convert(lth&x&vr&u,a,*,sum)) ge 1 then &&
                  convert(lth&x&vr&u,a,*,sum)/ &&
                  convert(lth&x&fp&u,a,*,sum) else &&
                  convert(lth&nrtot&vr&u,a,*,sum)/ &&
                  convert(lth&nrtot&fp&u,a,*,sum)) else &&
                 convert(work'id(var)&nrtot&vr&u,a,*,sum)/ &&
                 convert(work'id(var)&nrtot&fp&u,a,*,sum))

```

-- FASTE PRISER 95=100:

-- NB!! formelen må rettes når nytt grunnlag/basisåret endres

```

local new i&x = if t ge 78:1 and t le 78:4 then &&
                prod(p&x<date 79 to 95>)*l&x &&
                else if t ge 79:1 and t le 79:4 then &&
                prod(p&x<date 80 to 95>)*l&x &&
                else if t ge 80:1 and t le 80:4 then &&
                prod(p&x<date 81 to 95>)*l&x &&
                else if t ge 81:1 and t le 81:4 then &&
                prod(p&x<date 82 to 95>)*l&x &&
                else if t ge 82:1 and t le 82:4 then &&
                prod(p&x<date 83 to 95>)*l&x &&
                else if t ge 83:1 and t le 83:4 then &&
                prod(p&x<date 84 to 95>)*l&x &&
                else if t ge 84:1 and t le 84:4 then &&
                prod(p&x<date 85 to 95>)*l&x &&
                else if t ge 85:1 and t le 85:4 then &&
                prod(p&x<date 86 to 95>)*l&x &&
                else if t ge 86:1 and t le 86:4 then &&
                prod(p&x<date 87 to 95>)*l&x &&
                else if t ge 87:1 and t le 87:4 then &&
                prod(p&x<date 88 to 95>)*l&x &&
                else if t ge 88:1 and t le 88:4 then &&
                prod(p&x<date 89 to 95>)*l&x &&
                else if t ge 89:1 and t le 89:4 then &&
                prod(p&x<date 90 to 95>)*l&x &&
                else if t ge 90:1 and t le 90:4 then &&
                prod(p&x<date 91 to 95>)*l&x &&
                else if t ge 91:1 and t le 91:4 then &&
                prod(p&x<date 92 to 95>)*l&x &&
                else if t ge 92:1 and t le 92:4 then &&
                prod(p&x<date 93 to 95>)*l&x &&
                else if t ge 93:1 and t le 93:4 then &&
                prod(p&x<date 94 to 95>)*l&x &&
                else if t ge 94:1 and t le 94:4 then &&
                prod(p&x<date 95 to 95>)*l&x &&
                else if t ge 95:1 and t le 98:4 then &&
                f&x &&
                else ND

```

-- NIVÅTALL I FASTE OG LØPENDE PRISER:

```
id(var)&x&vr&u = l&x
```

```
id(var)&x&vl&u = if f&x eq 0 then 0 else i&x
```

```
if var eq "BIL" -- beregner ufordelt lager
```

```
ufv&nrtot&vr&u = ufv&nrtot&vr&u
```

```
ufv&nrtot&vl&u = utdat'ufv&nrtot&vr&u/ &&
```

```
(utdat'lth&nrtot&vr&u/utdat'lth&nrtot&vl&u)
```

```
end if
```

```
END LOOP
```



```

----- Årstall -----
else

if var eq "BIL" or var eq "LTH" -- beregner leveranse til hjemmemarkedet
local index=0
LOOP for x in nrlnrva
set index=index+1
if exists(bil&x&vr)
work'lth&x&vr = if not exists(prob&x&vr) then series(0) else &&
                overlay(prob&x&vr,series(0)) - &&
                (if not exists(eks&x&vr) then series(0) &&
                else overlay(eks&x&vr,series(0)))

work'lth&x&fp = if not exists(prob&x&fp) then series(0) else &&
                overlay(prob&x&fp,series(0)) - &&
                (if not exists(eks&x&fp) then series(0) &&
                else overlay(eks&x&fp,series(0)))

end if
END LOOP

-- Reisetrafikk (utlendingers konsum i Norge) har kun verdi for eksport
set work'lth.nr010.vr = series(0)
set work'lth.nr010.fp = series(0)

work'lth&nrtot&vr = lsum(selectnames(crosslist({lth},nrlnrva,{vr}), &&
exists(work'@name)))
work'lth&nrtot&fp = lsum(selectnames(crosslist({lth},nrlnrva,{fp}), &&
exists(work'@name)))
else
if agg eq "U" -- til totalindeksen (summerer opp ukjedede tall)
if not exists(work'id(var)&nrtot&vr)
work'id(var)&nrtot&vr = lsum(selectnames(crosslist({id(var)},liste, &&
{vr}),exists(@name)))
work'id(var)&nrtot&fp = lsum(selectnames(crosslist({id(var)},liste, &&
{fp}),exists(@name)))

end if
end if
end if

-- Oracle-databasen inneholder ikke serier som er lik null

local index=0
LOOP for x in liste
set index=index+1

if name(id(var)&x&vr) eq "BNPB.NR2958.VR" or && -- Disse seriene skal være lik
name(id(var)&x&vr) eq "PROB.NR2958.VR" -- null i løpende pri ser, men
-- ikke i faste

local new l&x = if not exists(id(var)&x&vr) then series(0) &&
                else overlay(id(var)&x&vr,series(0))
local new f&x = overlay(id(var)&x&fp,series(0))
local new p&x = work'id(var)&nrtot&vr/work'id(var)&nrtot&fp

-- FASTE PRISER 95=100 (NR2958):
-- NB!! formelen må rettes når nytt grunnlag/basisår endres

local new i&x = if t ge 78 and t le 79 then &&
                prod(p&x<date 79 to 95>)*f&x &&
                else if t eq 80 then &&
                prod(p&x<date 80 to 95>)*f&x &&
                else if t eq 81 then &&
                prod(p&x<date 81 to 95>)*f&x &&

```

```

else if t eq 82 then &&
  prod(p&x<date 82 to 95>)*f&x &&
else if t eq 83 then &&
  prod(p&x<date 83 to 95>)*f&x &&
else if t eq 84 then &&
  prod(p&x<date 84 to 95>)*f&x &&
else if t eq 85 then &&
  prod(p&x<date 85 to 95>)*f&x &&
else if t eq 86 then &&
  prod(p&x<date 86 to 95>)*f&x &&
else if t eq 87 then &&
  prod(p&x<date 87 to 95>)*f&x &&
else if t le 88 then &&
  prod(p&x<date 88 to 95>)*f&x &&
else if t eq 89 then &&
  prod(p&x<date 89 to 95>)*f&x &&
else if t eq 90 then &&
  prod(p&x<date 90 to 95>)*f&x &&
else if t eq 91 then &&
  prod(p&x<date 91 to 95>)*f&x &&
else if t eq 92 then &&
  prod(p&x<date 92 to 95>)*f&x &&
else if t eq 93 then &&
  prod(p&x<date 93 to 95>)*f&x &&
else if t eq 94 then &&
  prod(p&x<date 94 to 95>)*f&x &&
else if t eq 95 then &&
  l&x &&
  else if t ge 96 and t le 98 then &&
    f&x &&
  else ND

```

-- NIVÅTALL I FASTE OG LØPENDE PRISER (NR2958):

```

id(var)&x&vr = l&x
id(var)&x&vl = if f&x eq 0 then 0 else i&x

else

if agg eq "U" or agg eq "u"
local new l&x = if not exists(id(var)&x&vr) then series(0) &&
  else overlay(id(var)&x&vr,series(0))
local new f&x = if not exists(id(var)&x&fp) then series(0) &&
  else overlay(id(var)&x&fp,series(0))

else if agg eq "A" or agg eq "a"
if not exists(id(var)&x&vr)
  type "Aggregat mangler for "+name(id(var)&x)
  write "Aggregat mangler for "+name(id(var)&x) to feilmelding
  next
else
if var eq "DEP" or var eq "KAPN" or var eq "KAPB"
  if name(x) eq "NRMARK" or name(x) eq "NRIMARK"
    local new l&x = overlay(nrkap_agg'id(var)&x&vr,&&
      knrkap_agg'id(var)&x&vr)
    local new f&x = overlay(nrkap_agg'id(var)&x&fp,&&
      knrkap_agg'id(var)&x&fp)
  else
    local new l&x = id(var)&x&vr
    if exists(id(var)&x&fp)
      local new f&x = overlay(id(var)&x&fp,series(0))
    else
      local new f&x = series(0)

```

```

    end if
  end if
else
  if name(x) eq "NRMARK" or name(x) eq "NRIMARK"
    local new l&x = if exists(work'id(var)&x&vr) then work'id(var)&x&vr &&
      else nr02'id(var)&x&vr
    set <date basisaar+1 to slup> l&x = convert(id(var)&x&vr&u,a)
    local new f&x = if exists(work'id(var)&x&fp) then work'id(var)&x&fp &&
      else nr02'id(var)&x&fp
    set <date basisaar+1 to slup> f&x = convert(id(var)&x&vl&u,a)
  else
    local new l&x = overlay(id(var)&x&vr,series(0))
    local new f&x = overlay(id(var)&x&fp,series(0))
  end if
end if
end if
end if

if var eq "BIL" and (agg eq "U" or agg eq "u")
local new p&x = if work'lth&x&vr/work'lth&x&fp ne NC then &&
  work'lth&x&vr/work'lth&x&fp else work'lth&nr02&vr/work'lth&nr02&fp

else if var eq "BIL" and (agg eq "A" or agg eq "a")
local new p&x = work'lth&nr02&vr/work'lth&nr02&fp

else
  local new p&x = overlay(l&x/f&x,work'id(var)&nr02&vr/work'id(var)&nr02&fp)
end if

```

-- FASTE PRISER 95=100:

-- NB!! formelen må rettes når nytt grunnlag/basisåret endres

```

local new i&x = if t eq 78 then &&
  prod(p&x<date 79 to 95>)*1&x &&
  else if t eq 79 then &&
  prod(p&x<date 80 to 95>)*1&x &&
  else if t eq 80 then &&
  prod(p&x<date 81 to 95>)*1&x &&
  else if t eq 81 then &&
  prod(p&x<date 82 to 95>)*1&x &&
  else if t eq 82 then &&
  prod(p&x<date 83 to 95>)*1&x &&
  else if t eq 83 then &&
  prod(p&x<date 84 to 95>)*1&x &&
  else if t eq 84 then &&
  prod(p&x<date 85 to 95>)*1&x &&
  else if t eq 85 then &&
  prod(p&x<date 86 to 95>)*1&x &&
  else if t eq 86 then &&
  prod(p&x<date 87 to 95>)*1&x &&
  else if t eq 87 then &&
  prod(p&x<date 88 to 95>)*1&x &&
  else if t le 88 then &&
  prod(p&x<date 89 to 95>)*1&x &&
  else if t eq 89 then &&
  prod(p&x<date 90 to 95>)*1&x &&
  else if t eq 90 then &&
  prod(p&x<date 91 to 95>)*1&x &&
  else if t eq 91 then &&
  prod(p&x<date 92 to 95>)*1&x &&
  else if t eq 92 then &&
  prod(p&x<date 93 to 95>)*1&x &&
  else if t eq 93 then &&

```

```
prod(p&x<date 94 to 95>)*l&x &&  
else if t eq 94 then &&  
prod(p&x<date 95 to 95>)*l&x &&  
else if t eq 95 then &&  
l&x &&  
else if t ge 96 and t le 98 then &&  
f&x &&  
else ND
```

-- NIVÅTALL I FASTE OG LØPENDE PRISER:

```
id(var)&x&vr = l&x  
id(var)&x&vl = if l&x eq 0 then 0 else i&x
```

-- Pga. avstemming av KNR settes ufordelt lager lik null

```
if var eq "BIL" and (agg eq "U" or agg eq "u")  
  ufv&x&vr = series(0)  
  ufv&x&vl = series(0)  
end if
```

```
end if  
END LOOP
```

```
end if
```

```
end block  
END PROCEDURE
```

Vedlegg 3: Styreprogrammet \$velg_refaar

```
-- Programmert av: Pia E. Tønjum, 95/96
-- Sist endret: PET,17/7-98 (Rettet feil i blokkene)
-- Prosedyrene kan brukes til åpning av databaser (lesetillatelser) og valg
-- av referanseår, og sjekker evt. om gyldig referanseår er valgt. Referanseår
-- bestemmes av brukeren for hver gang formlene benyttes med kommandoen:
-- -/referanse_aar = 1990 (f.eks). Må være definert i tabell-progr/input-fil
-- Prosedyrene lager evt. formler for uttak av data med referanseår valgt av
-- brukeren. Formlene oppdateres bare dersom kontoplanen eller listene som
-- benyttes endres (f. eks. hvis nye aggregater) eller antall lister utvides
-- (flere prosedyrekall legges inn). Var.beskr. må i såfall legges inn på nytt
```

```
PROCEDURE $velg_refaar          -- styreprogram for omregningsformler
```

```
block
clear
over on
glue dot
close all
channel warning none
```

```
-- LASTER PROSEDYRENE
```

```
load "$NASJREGN/kjed/kjed_valgt_refaar"  -- lager omregningsformler
load "$NASJREGN/kjed/kjed"              -- bruker hjelpeprosedyrer
```

```
-- liste med alle blokkene som inngår i rutinene for valg av nytt referanseår
-alle_serier = {prod,bil,kons,inv,eks,imp,lth,dep,divagg}
```

```
type "-----"
type "BRUKER FERDIG KJEDEDE ÅRSTALL, OG LAGER FORMLER TIL"
type "BEREGNING AV NYE FASTPRISTALL FOR ÅR ELLER KVARTAL MED "
type "NYTT REFERANSEÅR DERSOM KONTOPLANEN ER ENDRET"
type blank
type "EVT. ÅPNES DATABASER MED LESETILLATELSER, OG REFERANSEÅR "
type "VELGES FOR UTTAK AV DATA"
type "-----"
type blank
type blank
```

```
response keyboard
scalar work'jn:string
accept "Er nr-db kjedet til grunnlagsårets (knr) priser (j/n) ":jn
```

```
if jn eq "n"
  type "Da må evt. databanken kjedes først (prosedyre $kjed) -"
  type "det må være gjort både for å lage/eksekvere formler!"
  return
else
  $gi_hjelp
end if
```

```
-- ÅRS- ELLER KVARTALSTALL/PERIODISITET
```

```
scalar work'frekvens: string
accept "Års- eller kvartalstall (a/q)? ":frekvens
```

```

if frekvens eq "q"
  freq q
else if frekvens eq "a"
  freq a
else
  type "Ugyldig periodisitet: " + string(frekvens)
  accept "Års- eller kvartalstall (a/q)? ":frekvens
end if

-----
-- AGGREGERINGSNIVÅ (KNR ELLER NR)
-----

scalar work'aggnivaa: string    -- knr/nr
scalar work'agg: string        -- aggregert/disaggregert (a/u)

if frekvens eq "a"
  accept "Årstill etter KNR- eller NR-nivå (knr/nr)?" :aggnivaa
  if not (aggnivaa eq "knr" or aggnivaa eq "KNR")
    if not (aggnivaa eq "nr" or aggnivaa eq "NR")
      type "Ugyldig aggregeringsnivå!"
      accept "Årstill etter KNR- eller NR-nivå (knr/nr)?" :aggnivaa
    end if
  end if
else
  -/aggnivaa = "knr"
end if

-----
-- LESETILLATELSER
-----

scalar offentlig: string

open <access read> "$NASJREGN/knr/struktur/kontoplan" as kplan

if frekvens eq "a"
  if aggnivaa eq "knr" or aggnivaa eq "KNR"
    open <acc read> "$NASJREGN/kjed/knrpub95" as nr
    scalar db: string = "velg_refaar_nr"
  else
    open <acc read> "$NASJREGN/kjed/nrpub95" as nrpub
    scalar db: string = "velg_refaar_nrpub"
  end if
else if frekvens eq "q"
  open <acc read> "$NASJREGN/knr/modell/knr95/knr95arb" as knr95arb
  open <acc read> "$NASJREGN/knr/modell/knr95/knr95korr_agg" as knr95korr_agg
  open <acc read> "$NASJREGN/kjed/knrpub95" as nr
  -- åpnes i alle tilfelle for å lage liste til loop gjennom hele db
  open <access read> "/ssb/frisch/fame/knr/statist" as statist
  scalar db: string = "velg_refaar_knr"
end if

accept "Skal formler oppdateres (j/n)? ":jn

if jn eq "n"
  open <access read> file(db) as velg_refaar
  type blank
  type "Nå er følgende databaser åpnet med lesetillatelse: "
  type @open.db
else

```

```
-- OPPRETTER NY, ÅPNER ELLER TØMMER EKSISTERENDE DB
```

```
accept "Skal alle gamle resultater slettes (j/n)? ":jn
if jn eq "n"
  open <access shar> file(db) as velg_refaar
else
  open <access over> file(db) as velg_refaar
end if
store velg_refaar
type blank
type "Nå er følgende databaser åpnet: "
type @open.db
type "Har åpnet databasen " +name(id(db))+ " som VELG_REFAAR"
type "Lagrer formler i " + @store
type blank
end if
```

```
-- BEREGNINGSPERIODE OG REFERANSEÅR
```

```
if aggnivaa eq "knr" or aggnivaa eq "KNR"
  scalar foerste_aar:date(annual) = firstvalue(bnpb.nr23_9.v1)
  scalar siste_aar:date(annual) = lastvalue(bnpb.nr23_9.v1)
else
  scalar foerste_aar:date(annual) = firstvalue(bnpb.nr2_.v1)
  scalar siste_aar:date(annual) = lastvalue(bnpb.nr2_.v1)
end if

scalar work'referanse_aar:date(annual)
accept "Nytt referanseår? ": referanse_aar

if not (referanse_aar ge foerste_aar and referanse_aar le siste_aar)
  type "Ulovlig referanseår!"
  type "Nasjonalregnstall finnes ikke for " + string(year(referanse_aar))
  type "Velg referanseår innenfor perioden " + &&
    string(year(foerste_aar)) + " - " + string(year(siste_aar))
  accept "Nytt referanseår? ": referanse_aar
end if

date foerste_aar to siste_aar

if @store eq "AUTO" -- skal bare lese fra databasene
  type blank
  type string(year(referanse_aar)) + " er valgt som referanseår. " + &&
    "Om du ønsker å skrive ut data med et annet referanseår, må du "+&&
    "velge referanseår innen intervallet : " + string(year(foerste_aar)) +&&
    " - "+ string(year(siste_aar)) + ", og gi f.eks. følgende kommando: "+&&
    "-/referanse_aar = 1992 "
  return
end if

accept "Lage formler for hele databanken (j/n)?" :jn
```

```
-- LOOP GJENNOM HELE DATABANKEN
```

```
-/ingen_liste={ }
```

```
if jn eq "j"
```

```

type "Lager først variabelliste (case-serie) med alle sektorer"
type "til looping gjennom hele databasen. Det tar litt tid ....."
type blank

series <store work; case 1 to *> knr_variabelliste : string by case
series <store work; case 1 to *> nr_variabelliste : string by case
series <store work; case 1 to *> nrpub_variabelliste : string by case
series <store work; case 1 to *> alle_knr : string by case
series <store work; case 1 to *> alle_nr : string by case
series <store work; case 1 to *> alle_nrpub : string by case

if frekvens eq "a"
  if aggnivaa eq "knr"
    local new i = 1
    LOOP for x in wildlist(nr,"?VL") except wildlist(nr,"?blank?")
      set alle_nr[i] = replace(string(name(x)),".VL","",last)
      set i = i+1
    END LOOP
    -/hele_db = nl(alle_nr)
  else
    local new i = 1
    LOOP for x in wildlist(nrpub,"?VL") except wildlist(nrpub,"?blank?")
      set alle_nrpub[i] = replace(string(name(x)),".VL","",last)
      set i = i+1
    END LOOP
    -/hele_db = nl(alle_nrpub)
  end if
else
  local new i = 1
  LOOP for x in wildlist(ostatist,"?VL.U") except wildlist(ostatist,"?blank?")
    set alle_knr[i] = replace(string(name(x)),".VL.U","",last)
    set i = i+1
  END LOOP
  -/hele_db = nl(alle_knr)
end if

type blank
type "LAGER OMREGNINGSFORMLER FOR ALLE SERIENE I DATABANKEN"

$kjed_valgt_refaar hele_db, ingen_liste, "ALLE"
$slag_blank

type blank
type "FORMLER FERDIG. NYTT REFERANSEÅR ER: " + string(year(referanse_aar))
type "Om du ønsker å skrive ut data med et annet referanseår, må du "+&&
  "velge referanseår innen intervallet : " + string(year(foerste_aar)) +&&
  " - " + string(year(siste_aar))+ " , og gi f.eks. følgende kommando: "+&&
  "-/referanse_aar = 1992 "
return
else
  $velg_refaar_en
end if
end block
END PROCEDURE -- $velg_refaar

-----
PROCEDURE $velg_refaar_en          -- kalles av $velg_refaar
-----

scalar work'variabel: string
accept "Variabel som skal kjedes? ":variabel
accept "Omregne aggregerte eller uaggregerte tall (a/u)? ":agg

```



```

if variabel eq "h"
  $gi_hjelp
else if (variabel eq "prod" or variabel eq "dep" or variabel eq "kons" or &&
variabel eq "inv" or variabel eq "eks" or variabel eq "imp" or &&
variabel eq "divagg" or variabel eq "bil" or variabel eq "lth" or &&
variabel eq "alle") and ((aggnivaa eq "knr") or &&
(aggnivaa eq "nr" and (agg eq "a" or agg eq "A")))
  -/prosedyre = "$velg_refaar_" + string(variabel)
  execute prosedyre
else
  type "Ugyldig valg: Variabel "+upper(string(variabel))
  type "Prøv igjen (tast 'h' for hjelp)"
  type blank
  $velg_refaar_en
end if

```

\$ferdig

END PROCEDURE -- \$velg_refaar_en

PROCEDURE \$velg_refaar_alle

LOOP for x in alle_serier

```

-/alle_variable = lower(name(x))
-/prosedyre = "$velg_refaar_"+alle_variable+"
execute prosedyre
END LOOP

```

END PROCEDURE -- \$velg_refaar_alle

PROCEDURE \$velg_refaar_prod

```

-- til kjeding av vareinnsats brukes nrlknrps29 istedenfor nrlknrps på
-- detaljert knr-nivå pga. avstemmingen, da knr-databanken inneholder
-- sektor nr2958

```

```

type "PRODUKSJONSSEKTORER, "+&&
upper(string(aggnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

```

```

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

```

```

$kjed_valgt_refaar nrlknrps29, ingen_liste, "PROB"
$kjed_valgt_refaar nrlknrva, ingen_liste, "PROB"
$kjed_valgt_refaar nrlknrps, ingen_liste, "PINE"
$kjed_valgt_refaar nrlknrps, ingen_liste, "PINF"
$kjed_valgt_refaar nrlknrps29, ingen_liste, "PIN"
$kjed_valgt_refaar nrlknrps29, ingen_liste, "BNPB"

```

```

else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

```

```

-/aggps=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggps.txt | grep -i 'nr"',case))
-/aggps29=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggps29.txt | grep -i 'nr"',case))
-/aggpin=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggpin.txt | grep -i 'nr"',case))
-/aggdivps = nl(system("grep -i 'bnpb.nr' "+&&
"$NASJREGN/knr/tabell/tab_makro.txt | cut -c 8-25",case)) except &&
{nr29korr} union {nr23_6,nrprim}

```

```

-/aggdivprim = {nr051_2,nrprim,nrprim,nrjord,nr23vare}
-/aggfindep=nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_bnpbfindep.txt | grep -i 'nr'",case))

$kjed_valgt_refaar aggps, ingen_liste, "PROB"
$kjed_valgt_refaar aggdivprim, ingen_liste, "PROB"
$kjed_valgt_refaar aggps29, ingen_liste, "PIN"
$kjed_valgt_refaar aggdivps, ingen_liste, "PIN"
$kjed_valgt_refaar aggps29, ingen_liste, "BNPB"
$kjed_valgt_refaar aggdivps, ingen_liste, "BNPB"
$kjed_valgt_refaar aggfindep, ingen_liste, "BNPB"

else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")

-/aggpubhprob = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubhov_prob.txt | grep -i 'nr'",case))
-/aggpubnprob = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubnaer_prob.txt | grep -i 'nr'",case))
-/aggpubhpin = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubhov_pin.txt | grep -i 'nr'",case))
-/aggpubnpin = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubnaer_pin.txt | grep -i 'nr'",case))
-/aggpubhbnpb = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubhov_bnpb.txt | grep -i 'nr'",case))
-/aggpubnbnpb = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubnaer.txt | grep -i 'nr'",case))

$kjed_valgt_refaar aggpubnprob, ingen_liste, "PROB"
$kjed_valgt_refaar aggpubhprob, ingen_liste, "PROB"
$kjed_valgt_refaar aggpubnpin, ingen_liste, "PIN"
$kjed_valgt_refaar aggpubhpin, ingen_liste, "PIN"
$kjed_valgt_refaar aggpubnbnpb, ingen_liste, "BNPB"
$kjed_valgt_refaar aggpubhbnpb, ingen_liste, "BNPB"

end if

type "Formler for produksjonssektorene er ferdig, kl. " + now
type blank

END PROCEDURE -- $velg_refaar_prod

-----
PROCEDURE $velg_refaar_kons
-----

type "KONSUM, "+&&
upper(string(aggnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

$kjed_valgt_refaar nrknrcp, ingen_liste, "KOH"
$kjed_valgt_refaar nrknrci, ingen_liste, "KOI"
$kjed_valgt_refaar nrknrg, ingen_liste, "KOO"

else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

-/aggcp=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggcp.txt | grep -i 'nr'",case)) union &&
{nr6151.nr0943}
-/aggkons=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggkons.txt | grep -i 'nr'",case)) union {nr66tjen}

$kjed_valgt_refaar aggcp, ingen_liste, "KOH"
$kjed_valgt_refaar aggkons, ingen_liste, "KONS"

```

```

else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")

-/agppubkoh = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubkoh.txt | grep -i 'nr'",case))
-/agppubkoi = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubkoi.txt | grep -i 'nr'",case))
-/agppubkoo = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubkoo.txt | grep -i 'nr'",case))
-/agppubkos=nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubkos.txt | grep -i 'nr'",case))
-/agppubkok=nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubkok.txt | grep -i 'nr'",case))
-/agpkons=nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_agpkons.txt | grep -i 'nr'",case))

$kjed_valgt_refaar agppubkoh, ingen_liste, "KOH"
$kjed_valgt_refaar agppubkoo, ingen_liste, "KOO"
$kjed_valgt_refaar agppubkoi, ingen_liste, "KOI"
$kjed_valgt_refaar agppubkos, ingen_liste, "KOO"
$kjed_valgt_refaar agppubkok, ingen_liste, "KOO"
$kjed_agg agpkons
end if

type "Formler for konsum ferdig, kl. " + now
type blank

END PROCEDURE -- $velg_refaar_kons

-----
PROCEDURE $velg_refaar_eks
-----

type "EKSPORT, "+&&
upper(string(aggnivaa))+".nivå, aggregeringsnivå: " + upper(string(agg))

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

-/nrlnrjr09= {nr0915,nr0937,nr0938,nr0939,nr0941,nr0942,nr0943, &&
nr0955,nr0971 }

$kjed_valgt_refaar nrlnrva, ingen_liste, "EKS"
$kjed_valgt_refaar nrlnrjr09, ingen_liste, "EKS"
$kjed_valgt_refaar nrlnrva, ingen_liste, "EKSB"
$kjed_valgt_refaar nrlnrjr09, ingen_liste, "EKSB"

else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

-/aggeks=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggeks.txt | grep -i 'nr'",case))
-/aggeksb = {nrtot,nr051_2,nrprimp,nrjord}

$kjed_valgt_refaar aggeks, ingen_liste, "EKS"
$kjed_valgt_refaar aggeksb, ingen_liste, "EKSB"

else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")

series <store work; case 1 to *> alle_eks : string by case

-/i = 1
LOOP for x in wildlist(nrpub,"eks?vl")
set alle_eks[i] = replace(replace(string(name(x)),".VL","",last),"EKS.", "",first)
set i = i+1
END LOOP

```

```

-/aggeks = nl(alle_eks)

$skjed_valgt_refaar aggeks, ingen_liste, "EKS"

end if

type "Formler for eksport er ferdig, kl. " + now
type blank

END PROCEDURE -- $velg_refaar_eks

-----
PROCEDURE $velg_refaar_imp
-----

type "IMPORT, "+&&
upper(string(aggnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

$skjed_valgt_refaar nrlknrva, ingen_liste, "IMP"

else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

-/aggimp=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggimp.txt | grep -i 'nr",case)) union {nr051_2,nrprimp,nrjord}

$skjed_valgt_refaar aggimp, ingen_liste, "IMP"

else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")

series <store work; case 1 to *> alle_imp : string by case

-/i = 1
LOOP for x in wildlist(nrpub,"imp?vl")
    set alle_imp[i] = replace(replace(string(name(x)), ".VL", "", ,last), "IMP.", "", ,first)
    set i = i+1
END LOOP

-/aggimp = nl(alle_imp)

$skjed_valgt_refaar aggimp, ingen_liste, "IMP"

end if

type "Formler for import er ferdig, kl. " + now
type blank

END PROCEDURE -- velg_refaar_imp

-----
PROCEDURE $velg_refaar_inv
-----

type "INVESTERINGSSEKTOR OG INVESTERINGSEKTOR*ART, "
type upper(string(aggnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

$skjed_valgt_refaar nrlknrjs, ingen_liste, "SBRK"
$skjed_valgt_refaar nrlknrjr, ingen_liste, "SBRK"
$skjed_valgt_refaar nrlknrjs, nrlknrjr, "SBRK"

type "Formler for SBRK ferdig, kl. " + now

```

```
$skjed_valgt_refaar nrlknrjs, ingen_liste, "NYI"
$skjed_valgt_refaar nrlknrjr, ingen_liste, "NYI"
$skjed_valgt_refaar nrlknrjs, nrlknrjr, "NYI"
```

```
type "Formler for NYI er ferdig, kl. " + now
```

```
$skjed_valgt_refaar nrlknrjs, ingen_liste, "BIF"
$skjed_valgt_refaar nrlknrjr, ingen_liste, "BIF"
$skjed_valgt_refaar nrlknrjs, nrlknrjr, "BIF"
```

```
type "Formler for BIF er ferdig, kl. " + now
```

```
else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")
```

```
-/aggjbm = {nr08b,nr08m}
-/aggjs=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggjs.txt | grep -i 'nr' ",case)) union &&
{nrprim} except {nr8368}
-/aggjsjr=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggjsjr.txt | grep -i 'nr' ",case)) union &&
{nrprim} except {nr8368}
-/aggdivjs = nl(system("grep -i 'bif.nr' "+&&
"$NASJREGN/knr/tabell/tab_makro.txt | cut -c 7-25",case)) union &&
{nr830ljs,nr83_,nr86_,nrimark,nrmark} except {nr8_} union aggjbm
```

```
$skjed_valgt_refaar aggjsjr, ingen_liste, "SBRK"
$skjed_valgt_refaar nrlknrjs, aggjbm, "SBRK"
$skjed_valgt_refaar aggjs, aggjbm, "SBRK"
$skjed_valgt_refaar aggdivjs, ingen_liste, "SBRK"
$skjed_valgt_refaar aggdivjs, aggjbm, "SBRK"
```

```
type "Formler for SBRK er ferdig, kl. " + now
```

```
$skjed_valgt_refaar aggjsjr, ingen_liste, "NYI"
$skjed_valgt_refaar nrlknrjs, aggjbm, "NYI"
$skjed_valgt_refaar aggjs, aggjbm, "NYI"
$skjed_valgt_refaar aggdivjs, ingen_liste, "NYI"
$skjed_valgt_refaar aggdivjs, aggjbm, "NYI"
```

```
type "Formler for NYI er ferdig, kl. " + now
```

```
$skjed_valgt_refaar aggjsjr, ingen_liste, "BIF"
$skjed_valgt_refaar nrlknrjs, aggjbm, "BIF"
$skjed_valgt_refaar aggjs, aggjbm, "BIF"
$skjed_valgt_refaar aggdivjs, ingen_liste, "BIF"
$skjed_valgt_refaar aggdivjs, aggjbm, "BIF"
```

```
type "Formler for BIF er ferdig, kl. " + now
```

```
else if aggnivaa eq "nr" and (agg eq "u" or agg eq "U")
```

```
-/nrm = system("grep -i '^8' $NASJREGN/tidsserier/meta/mottaker.txt", case)
-/nrljs = nl( "NR" + substring(nrm,1, location(nrm, " ")-1) )
-/nrp = system("egrep -i '^008|^009' "+&&
"$NASJREGN/tidsserier/meta/produkt.txt", case)
-/nrljr = nl( "NR" + substring(nrp,1, location(nrp, " ")-1) )
```

```
$skjed_valgt_refaar nrljs, nrljr, "BIF"
```

```
type "Formler for BIF er ferdig, kl. " + now
```

```
else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")
```

```
-/agggpubinv = nl(system("cut -c 3-15 "+&&
```

```

"$NASJREGN/tidsserier/tabell/tab_pubinv.txt | grep -i 'nr' ",case))
-/aggpubinvart = nl(system("cut -c 3-15 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubinvart.txt | grep -i 'nr' ",case))

$kjed_valgt_refaar aggpubinv, ingen_liste, "BIF"
$kjed_valgt_refaar aggpubinvart, ingen_liste, "BIF"

type "Formler for BIF er ferdig, kl. " + now

end if

type "Formler for alle inv.sektorer/arter er ferdig, kl. " + now
type blank

END PROCEDURE -- $velg_refaar_inv

-----
PROCEDURE $velg_refaar_dep
-----

type "KAPITALSLIT ETTER SEKTOR OG SEKTOR*ART, "+&&
upper(string(agnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")
  if frekvens eq "q"
    -/nrknrps=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggps24_5.txt | grep -i 'nr' ",case))
  end if

  $kjed_valgt_refaar nrknrps, nrknrjr, "DEP"
  $kjed_valgt_refaar nrknrps, ingen_liste, "DEP"

else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

  -/aggjbm = {nr08b,nr08m}
  -/aggps=nl(system("cut -c 3-15 "+&&
"$NASJREGN/knr/tabell/tab_aggps.txt | grep -i 'nr' ",case)) union &&
  {nrmark,nrmark}

  $kjed_valgt_refaar aggps, ingen_liste, "DEP"
  --$kjed_valgt_refaar nrknrps, aggjbm, "DEP" -- må evt. lage aggregater
  --$kjed_valgt_refaar aggps, aggjbm, "DEP"
  $kjed_valgt_refaar aggps, ingen_liste, "KAPB"
  $kjed_valgt_refaar aggps, ingen_liste, "KAPN"

end if

type "Formler for kapital slit er ferdig, kl. " + now
type blank

END PROCEDURE -- $velg_refaar_dep

-----
PROCEDURE $velg_refaar_lth
-----

type "LEVERANSE TIL HJEMMEMARKEDET, "+&&
upper(string(agnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

series <store work; case 1 to *> alle_lth : string by case

if aggnivaa ne "nr"

```

```

-i = 1
LOOP for x in wildlist(nr,"lth?vl")
    set alle_lth[i] = replace(replace(string(name(x)), ".VL", "", ,last), "LTH.", "", ,first)
    set i = i+1
END LOOP

-/agglth = nl(alle_lth) except nrlnrva
end if

if frekvens eq "a" and aggnivaa eq "nr"
    type "Leveranse til hjemmemarkedet beregnes ikke i årsregnskapet"
    type blank
else
    if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

        $skjed_valgt_refaar nrlnrva, ingen_liste, "LTH"

    else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

        $skjed_valgt_refaar agglth, ingen_liste, "LTH"

    end if

type "Formler for leveranse til hjemmemarkedet ferdig, kl. " + now
type blank

end if

END PROCEDURE -- $velg_refaar_lth

```

```

PROCEDURE $velg_refaar_bil

```

```

type "LAGERENDRING, UFORDELT LAGER, "+&&
upper(string(aggnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))

if aggnivaa eq "nr"
    type "Prosedyre kall er ikke lagt inn enda!"
    type blank
else
    if aggnivaa eq "knr" and (agg eq "u" or agg eq "U")

        $skjed_valgt_refaar nrlnrva, ingen_liste, "BIL"

    else if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")

        -/aggdivbil = {nr8lu_,nr8l_nr051_2,nrprimp,nrjord}

        $skjed_valgt_refaar aggdivbil, ingen_liste, "BIL"

    end if

type "Formler for lagerendring og ufordelt lager ferdig, kl. " + now
type blank

end if

END PROCEDURE -- $velg_refaar_bil

```

```
PROCEDURE $velg_refaar_divagg
```

```
type "DIV. AGGREGATER, "+&&
upper(string(aggnivaa))+"-nivå, aggregeringsnivå: " + upper(string(agg))
```

```
if agg eq "u" or agg eq "U"
  type "Finnes bare på agg-nivå 'a'"
  type blank
```

```
else
  if aggnivaa eq "knr" and (agg eq "a" or agg eq "A")
```

```
  -/aggmakro = nl(system("cut -c 3-20 "+&&
"$NASJREGN/knr/tabell/tab_makro.txt |grep -i 'nr' ",case))
  -/agganvendelse = nl(system("cut -c 3-20 "+&&
"$NASJREGN/knr/tabell/tab_anvendelse.txt | grep -i 'nr' ",case))
```

```
$skjed_valgt_refaar aggmakro, ingen_liste, "MAKROK"
$skjed_valgt_refaar agganvendelse, ingen_liste, "BNPANV"
```

```
type "Formler for makroøk. hovedstørrelser er ferdig, kl. " + now
type "Formler for BNP etter anvendelse er ferdig, kl. " + now
type blank
```

```
else if aggnivaa eq "nr" and (agg eq "a" or agg eq "A")
```

```
  -/aggpubtilg = nl(system("cut -c 3-20 "+&&
"$NASJREGN/tidsserier/tabell/tab_pubtilg.txt | grep -i 'nr' ",case)) &&
  union {tot.tilgialt,tot.anvialt}
```

```
$skjed_valgt_refaar aggpuntilg, ingen_liste, "TILGANV"
```

```
type "Formler for tilgang og anvendelse i alt er ferdig, kl. " + now
type blank
```

```
end if
end if
```

```
END PROCEDURE -- $velg_refaar_divagg
```


Vedlegg 4: Omregningsformler for valgfritt referanseår \$kjed_valgt_refaar

```
-- Programmert av Pia E.Tønjum, 95/96
-- Sist endret: PET, 25/10-96
-- Leser fra database som på forhånd er kjedet om til felles referanseår
-- (grunnlagsåret i knr) av $kjed, og lager omregningsformler for års- og
-- kvartalstall i faste priser med nytt (valgfritt) referanseår
```

```
-----
PROCEDURE $kjed_valgt_refaar      -- kalles av $velg_refaar
-----
```

```
argument liste1, &&          -- prduksjonssektor eller produktliste
       liste2, &&           -- artsliste, evt. en 'tom' liste
       var                  -- variabelnavn, f.eks. PROD, BNPB
```

```
block
over on
glue dot
```

```
----- Kvartalstall -----
if frekvens eq "q"
```

```
if name(liste1) eq "AGGANVENDELSE" or name(liste1) eq "AGGMAKRO" &&
or name(liste1) eq "AGGKONS" or name(liste1) eq "HELE_DB"
```

```
LOOP for y in crosstlist(liste1)
```

```
  -/x=name(y)
  -/vari=substring(name(y),1,location(name(y),".")-1)
```

```
  if exists(id("'" + x + ".vr"))
    execute "formula " + x + ".vlr.u = if " + x + ".vl.u eq 0 then 0 "+&&
           "else " + x + ".vl.u* (if " + x + ".vr[referanse_aar] "+&&
           "eq 0 then "+var+" .nrtot.vr[referanse_aar]/4/ "+&&
           "ave((" + var + ".nrtot.vl/4)<date referanse_aar>) "+&&
           "else " + x + ".vr[referanse_aar]/4/"+&&
           "ave((" + x + ".vl/4)<date referanse_aar>)"
```

```
  end if
END LOOP
```

```
else if name(liste2) ne "INGEN_LISTE"
```

```
LOOP for y in crosstlist({id("'" + var + "'")},liste1,liste2)
```

```
  -/x=name(y)

  if exists(id("'" + x + ".vr"))
    execute "formula " + x + ".vlr.u = if " + x + ".vl.u eq 0 then 0 "+&&
           "else " + x + ".vl.u* (if " + x + ".vr[referanse_aar] "+&&
           "eq 0 then "+var+" .nrtot.vr[referanse_aar]/4/ "+&&
           "ave((" + var + ".nrtot.vl/4)<date referanse_aar>) "+&&
           "else " + x + ".vr[referanse_aar]/4/"+&&
           "ave((" + x + ".vl/4)<date referanse_aar>)"
```

```
  end if
END LOOP
else
```

```
LOOP for y in crosstlist({id("'" + var + "'")},liste1)
```

```
  -/x=name(y)

  if exists(id("'" + x + ".vr"))
    execute "formula " + x + ".vlr.u = if " + x + ".vl.u eq 0 then 0 "+&&
           "else " + x + ".vl.u* (if " + x + ".vr[referanse_aar] "+&&
           "eq 0 then "+var+" .nrtot.vr[referanse_aar]/4/ "+&&
           "ave((" + var + ".nrtot.vl/4)<date referanse_aar>) "+&&
           "else " + x + ".vr[referanse_aar]/4/"+&&
           "ave((" + x + ".vl/4)<date referanse_aar>)"
```

```
  end if
```

```

END LOOP
end if

```

```

----- Årstill -----
else

```

```

if name(liste1) eq "AGGANVENDELSE" or name(liste1) eq "AGGMAKRO" &&
or name(liste1) eq "AGGKONS" or name(liste1) eq "AGGPUBTILG" &&
or name(liste1) eq "HELE_DB"

```

```

LOOP for y in crosstlist(liste1)
-/x=name(y)
-/vari=substring(name(y),1,location(name(y),".")-1)

```

```

if exists(id(""+x+ ".vr"))
execute "formula "+x+ ".v1r = if "+x+ ".v1 eq 0 then 0 "+&&
"else "+x+ ".v1* (if "+x+ ".vr[referanse_aar] "+&&
"eq 0 then "+vari+ ".nrtot.vr[referanse_aar]/ "+&&
""+vari+ ".nrtot.vl[referanse_aar] else "+&&
""+x+ ".vr[referanse_aar]/"+x+ ".v1[referanse_aar])"

```

```

end if
END LOOP

```

```

else if name(liste2) ne "INGEN_LISTE"

```

```

LOOP for y in crosstlist({id(""+var+ ""}),liste1,liste2)
-/x=name(y)

```

```

if exists(id(""+x+ ".vr"))
execute "formula "+x+ ".v1r = if "+x+ ".v1 eq 0 then 0 "+&&
"else "+x+ ".v1* (if "+x+ ".vr[referanse_aar] "+&&
"eq 0 then "+var+ ".nrtot.vr[referanse_aar]/ "+&&
""+var+ ".nrtot.vl[referanse_aar] else "+&&
""+x+ ".vr[referanse_aar]/"+x+ ".v1[referanse_aar])"

```

```

end if

```

```

END LOOP

```

```

else
LOOP for y in crosstlist({id(""+var+ ""}),liste1)
-/x=name(y)

```

```

if exists(id(""+x+ ".vr"))
execute "formula "+x+ ".v1r = if "+x+ ".v1 eq 0 then 0 "+&&
"else "+x+ ".v1* (if "+x+ ".vr[referanse_aar] "+&&
"eq 0 then "+var+ ".nrtot.vr[referanse_aar]/ "+&&
""+var+ ".nrtot.vl[referanse_aar] else "+&&
""+x+ ".vr[referanse_aar]/"+x+ ".v1[referanse_aar])"

```

```

end if
END LOOP

```

```

end if
end if

```

```

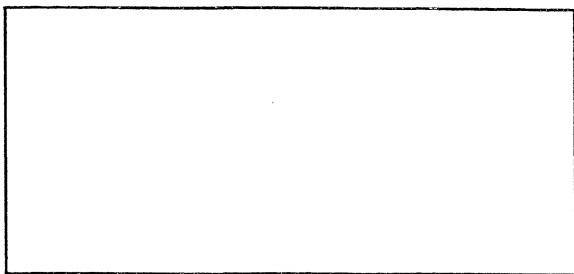
end block
END PROCEDURE

```

De sist utgitte publikasjonene i serien Notater

- 98/88 J. Sexton: Fremskrivning av tidsserier i KNR. 20s.
- 98/89 A.H. Foss: Definisjoner og beregningsmetoder for dødelighetstabell. 16s.
- 98/90 T. Dale: Samordnet levekårsundersøkelse 1998 - panelundersøkelsen: Dokumentasjonsrapport. 95s.
- 98/91 L. Lindholt: Rammvilkår for energigjenvinning av plast. 14s.
- 98/92 T. Vogt: Folatkunnskap blant kvinner i fertil alder: Dokumentasjonsrapport. 17s.
- 98/93 B.L. Western: Beregning av vektorer til inntekts- og formuesundersøkelsene 1995. 14s.
- 98/95 R. Johannessen: Prisindeks for hotellovernatting - delundersøkelse i konsumprisindeksen. 18s.
- 98/96 K.J. Einarsen: Definisjonskatalog for videregående opplæring: Utarbeidet av arbeidsutvalget i FylkesKOSTRA-utdanning og statistikkgruppen i Prosjekt LINDA-opplæring. 1. Utgave. 27s.
- 98/97 K.A. Brekke: Om metoder for beregning av miljøprofil for ulike varer, og hva vi trenger det til. 20s.
- 98/98 I.S. Wold: Modellering av husholdningenes transportkonsum for en analyse av grønne skatter: Muligheter og problemer innenfor rammen av en nyttetremodell. 81s.
- 98/99 R. Gudem: Utvikling av statistikk over bygg- og anleggsavfall. 56s.
- 98/100 T. Skjerpen: Konsumfordelingssystemet i KVARTS: Teknisk dokumentasjon. 42s.
- 99/1 F. Thorkildsen: FoB2000. Kobling av adresser fra GAB og DFS - status og utvikling. 22s.
- 99/2 K. Ibenholt og K.A. Brekke: Rammvilkår for produksjon av brunt papir. 12s.
- 99/3 I. Øyangen: Liv og helse i Akershus: Dokumentasjonsrapport. 22s.
- 99/4 M.V. Dysterud og E. Engeliën: Tettstedsavgrensning 1998: Teknisk dokumentasjon av nye rutiner. 53s.
- 99/5 Samordnede levekårsundersøkelser: Rapport fra brukerseminar 22. april 1996 og høringsuttalelser. 46s.
- 99/6 H. Utne: FoB2000. Sektordata: Statusrapport og tiltaksplan. 86s.
- 99/7 M. Sjøberg: Instruksjoner til og data frå eksperiment om internasjonal kvotehandel. 27s.
- 99/8 A. Sundvoll: Undersøkelse om kosthold blant spedbarn. 31s.
- 99/9 A. Schjalm: Sluttrapport om utvalg og estimering for kulturlandskapsovervåking. 16s.
- 99/10 E. Vinju: Statistikk over avfall og gjenvinning i industrien - 1996: Dokumentasjon av metode. 61s.
- 99/11 I. Aukrust og H. Utne: Detaljerte arbeidsmarkedsdata – på kommunenivå: Dokumentasjon av arbeidsmarkedsdata til PANDA. 4. utgave. 37s.
- 99/12 K.J. Einarsen, H. Hartvedt, A.B. Skara og C.B. Strand: Faktaark for FylkesKOSTRA-utdanning. Årsrapporteringen for 1998. Sør-Trøndelag fylkeskommune: Nøkkeltall med indikatorer for: Prioriteringer og ressursbruk, Dekningsgrad, Produktivitet og kvalitet. 41s.
- 99/13 G. Sparby: Selvangivelsesstatistikk 1993-1996: Dokumentasjon. 121s.
- 99/14 B.R. Joneid og J. Lajord: FD – Trygd: Dokumentasjonsrapport: Demografi. 1992-1993. 95s.
- 99/15 A.L. Brathaug: Evaluering av fylkesKOSTRA, helsetjenester 1998. 64s.
- 99/16 A.J. Isachsen, S.O. Stoknes og G.H. Bjønnes: Den store gjettekonkurransen. 26s.

Notater



Tillatelse nr.
159 000/502

B *Returadresse:*
Statistisk sentralbyrå
Postboks 8131 Dep.
N-0033 Oslo

Statistisk sentralbyrå

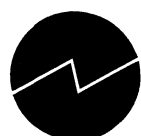
Oslo:
Postboks 8131 Dep.
0033 Oslo

Telefon: 22 86 45 00
Telefaks: 22 86 49 73

Kongsvinger:
Postboks 1260
2201 Kongsvinger

Telefon: 62 88 50 00
Telefaks: 62 88 50 30

ISSN 0806-3745



Statistisk sentralbyrå
Statistics Norway