

Institut for Folkesundhedsvidenskab
Biostatistisk Afdeling
Københavns Universitet

Introduktion til Regneøvelser i epidemiologi med SAS

Bendix Carstensen
Steno Diabetes Center
&
Biostatistisk afdeling
Institut for Folkesundhedsvidenskab
Københavns Universitet
bxc@steno.dk

Per Kragh Andersen
Biostatistisk afdeling
Institut for Folkesundhedsvidenskab
Københavns Universitet
p.k.andersen@biostat.ku.dk

Contents

I	Introduktion til SAS	1
1	SAS i relation til andre programpakker	1
2	Kald af SAS	1
2.1	Menu-styring: SAS Analyst	1
3	Programmering i SAS	2
4	Hvordan ser et SAS-program ud?	2
5	Datahåndtering, data-step	2
6	Procedurekald, proc-step	3
7	De basale procedurer	3
7.1	Udskrivning, proc print	3
7.2	Oversigt over datasæt, proc contents	4
7.3	Sortering, proc sort	4
7.4	Tabellering, proc freq	4
7.5	Udregning af basale stikprøvestørrelser, proc means og proc univariate	4
8	Sammenklustering af datasæt	5
8.1	Datasæt i forlængelse af hinanden	5
8.2	Datasæt ved siden af hinanden	5
8.3	Datasæt ved siden af hinanden — matchet	5
9	Datoer i SAS	6
9.1	Indlæsning af datoer	6
9.2	Udskrivning af datoer.	7
9.3	Faste datoer.	7
9.4	Regning med datoer.	7
9.5	Opgaver i dato-behandling	8
II	Follow-up data i SAS	9
10	Repræsentation af follow-up data	9
10.1	Gentagne events	9
11	Opdeling af risikotid	10
11.1	Eksempel	11
12	SMR-analyser (PYRS-udregninger)	13
III	Analyse af epidemiologiske data i SAS	15
13	proc freq	15
13.1	Tabellerede data	16
13.2	Stratificeret analyse	18
14	proc genmod	20
14.1	Analyse af kohortestudier	21
14.2	Analyse af case-kontrol studier	22

15	Udregning af odds-ratio, rate ratio og konfidensintervaller	25
15.1	estimate-statement i proc genmod	26
IV	Regneøvelser med SAS	29
1	The Framingham study	29
2	IHD data from Clayton & Hills.	30
3	Case-control study of BCG vaccination and leprosy.	31
4	Case-control study of malignant melanoma.	31
5	Testicular cancer risk and maternal parity.	33

Part I

Introduktion til SAS

1 SAS i relation til andre programpakker

Ved planlægningen af epidemiologikurser er vi i en valgsituation m.h.t valg af programmel. Der findes på markedet mange forskellige programpakker, der kan meget nær det samme, og ved valget af SAS vil vi ikke signalere, at andre muligheder er inferiøre.

Sammenlignes SAS med andre større kommercielle programpakker, som f.eks. Stata og SPSS, er der ikke væsentlige forskelle m.h.t. faciliteter, dog er Poisson regressionsanalyse af rater ret besværligt i SPSS. Om SAS kan siges:

- Fordele:
 1. Kan klare store datamængder.
 2. Har meget alsidige datahåndteringsfaciliteter.
 3. Kan give et højt niveau af dokumentation for datamanipulation og analyser med forholdsvis begrænset indsats.
 4. Kan udføre langt de fleste statistiske analyser.
 5. Fungerer på mange platforme.
 6. Fakultetet har en universallicens.
 7. Fås gratis, når man er ph.d. studerende ved fakultetet.
- Ulemper:
 1. Kan virke tungt at anvende da det er omfattende.
 2. Manualerne kan synes overvældene (manualen for `data`-steppet er alene på over 1000 sider!), og svære at finde rundt i.
 3. Selv mådelig grafik kræver omfattende programmering.

2 Kald af SAS

SAS kaldes fra Windows ved at klikke på SAS-ikonen. Derved får man tre vinduer frem på skærmen:

program editor-vindue— aktiveres med **F5**. Hvis flere editor-vinduer er åbne vil **F5** skifte mellem de forskellige editor-vinduer.

log-vindue— aktiveres med **F6**

output-vindue— aktiveres med **F7**

De øvrige funktionstasters funktion kan man få frem i et vindue på skærmen ved at trykke på **F9**.

Man kan lukke et vindue ved at trykke **Ctrl+F4**. (Pas på: **Alt+F4** vil lukke hele SAS).

2.1 Menu-styring: SAS Analyst

Der er to mulige måder at anvende SAS: Menu-styret eller program-styret. Menuerne kan man få frem ved at trykke **Solutions** → **Analysis** → **Analyst**, hvorefter man får adgang til et spreadsheet lignende interface hvor man kan taste data ind. Samtidig får man muligheder for at hente data ind fra forskellige typer af filer, såvel sædvanlige tekstfiler som SAS-systemfiler.

SAS Analyst indeholder en del peg-og-klik faciliteter til at lave forskellige statistiske analyser. Princippet er som i andre lignende systemer at man med diverse menuer får dannet et SAS-program som man så udfører. Dette vil være tilgængeligt således at man senere kan reproducere sine analyser.

Imidlertid indeholder Analyst ikke på nuværende tilpunkt noget interface til `proc genmod` som er helt central for epidemiologiske analyser.

3 Programmering i SAS

I *program editor*-vinduet kan man skrive et SAS-program. Vinduet opfører sig stort set som en hvilken som helst anden Windows-baseret editor hvad angår klip og klistring af tekst.

Et program kan f.eks. være:

```
data a ;
  input sex index blodtr ;
cards;
1 1.31 130
1 1.31 148
2 1.19 146
2 1.11 122
;
run ;
```

som udføres ved at trykke på **F8** eller **F3**. Hvis en del af *program editor*-vinduet markeres ved at holde **Shift** nede mens piletasterne bruges (eller med musen), vil kun den markerede del af programmet blive udført.

I *log*-vinduet vil en kopi af programmet flintre forbi sammen med noter om hvad der er blevet udført *samt* eventuelle fejlmeddelelser. Samtidig forsvinder teksten fra *program editor*-vinduet, og man ender i bunden af *output*-vinduet (hvis programmet producerer noget output).

Når man har skrevet et program er det fornuftigt at gemme det på disken ved **File**→**Save as**, eller **Ctrl-S**.

Det er en god ide at gemme dem under et letgennemskeligt navn, f.eks. **prj_name.sas** — det er en konvention at SAS-programmer har extension **.sas**. Så kan det senere hentes ind igen og evt. modificeres. Man kan dog altid hente det senest udførte program ind i *program editor*-vinduet ved at trykke på **F4**. (Pas på — hvis man trykker flere gange på **F4** bliver flere kopier af programmet hentet ind og sat efter hinanden).

4 Hvordan ser et SAS-program ud?

SAS-programmer består groft sagt af et antal s.k. “step”, som er af en af to typer:

- Datahåndtering, i **data**-step. Her indlæses man sine tal, definerer nye variable, f.eks. ved logaritmetransformation eller sammenlægning.

Et **data**-step er “indrammet” af:

```
data a ;
...
run ;
```

- Procedurekald, herunder grafik, **proc**-step. Når data foreligger på den rigtige facon, kan analyserne udføres ved hjælp af forskellige SAS-procedurer. (Der findes også SAS-procedurer til mere eksotiske former for datahåndtering).

Et **proc**-step er “indrammet” af:

```
proc xx data=a ;
...
run ;
```

5 Datahåndtering, data-step

Hvert statement i SAS afsluttes med et “;”. Der kan være flere statements på hver linje, eller et statement kan strække sig over flere linjer, og der kan være et vilkårligt antal blanktegn hvorsomhelst.

Det regnes for god tone højst at skrive et statement pr. linje, samt at afslutte alle **data**- og **proc**-step med **run**; og en blank linje. Derved muliggøres at programmet kan læses af andre samt af en selv en anden dag. Eksemplerne i SAS-manualerne følger disse konventioner.

Nedenfor er vist nogle eksempler på definition af nye variable, som forhåbentligvis er selvforklarende. Bemærk, at den naturlige logaritme betegnes med `log` i SAS, medens 10-tals logaritmen benævnes `log10`, og potensopløftning med `**`.

```
data b ;
  set a ;
  if ( sexnr eq 1 ) then sex = 'male' ;
  if ( sexnr eq 2 ) then sex = 'fem' ;
  logbp = log ( bp ) ;
  sqrtbp = sqrt ( bp ) ;
  chi2 = x ** 2 ;
  v8 = ( v3 gt 17 ) ;
  v9 = ( v3 gt 17 ) + ( v3 gt 24 ) ;
run ;
```

`data b` ; betyder: Nu dannes SAS-datasættet `b`.

`set a` ; betyder: Indlæs datasættet `a` og brug det som grundlag. Det vil sige at alle variable i `a` overføres til `b`. Når en variabel specificeres på venstre side af lighedstegnet bliver den automatisk tilføjet til datasættet `b`.

Bemærk de relationelle operatorer:

`eq` — equal to, lig med, kan også skrives `=`.

`ne` — not equal to, kan også skrives `^=`

`gt` — greater than, (skarpt) større end, kan også skrives `>`.

`ge` — greater than or equal to, større end eller lig med, kan også skrives `>=`.

`lt` — less than, (skarpt) mindre end, kan også skrives `<`.

`le` — less than or equal to, mindre end eller lig med, kan også skrives `<=`.

SAS har den konvention at et logisk udtryk som f.eks. `(v3 gt 17)` er 1 hvis det er sandt, og 0 ellers. I eksemplet ovenfor er `v8` 0 hvis `v3` er mindre end eller lig med 17, og 1 hvis den er større, mens `v9` er 0 hvis `v3` er mindre end eller lig med 17, 1 hvis `v3` er større end 17 og mindre end eller lig med 24, og 2 hvis `v3` er større end 24.

Der er SAS adgang til et meget stort antal funktioner, der alle er nærmere beskrevet i manualen, og i help-menuen [Help](#) → [SAS System Help](#) → [Contents](#) → [Help on SAS Software products](#) → [Base SAS Software](#) → [Using Base SAS Software](#) → [Working with the SAS Language](#) → [SAS Functions](#).

6 Procedurekald, proc-step

Når vi taler om de indbyggede SAS procedurer, er det bekvemt at skelne mellem 3 forskellige slags, nemlig

1. Basale procedurer til udskrivning, sortering og udregning af basale størrelser som gennemsnit og spredning mv.
2. Procedurer til egentlige statistiske analyser.
3. Procedurer til grafik.

7 De basale procedurer

7.1 Udskrivning, proc print

```
proc print data = sasuser.bp ;
  var sex bp ;
run ;
```

I linierne ovenfor behøver man kun at skrive `proc print;`. Det senest dannede datasæt vil da blive udskrevet i sin helhed.

Alle procedurer opfører sig på denne måde, men det er god programmeringsskik at skrive navnet på datasættet med hver gang, idet man derved undgår overraskelser når man laver lidt længere SAS-programmer hvor flere datasæt optræder imellem hinanden.

Man kan som ovenfor angive hvilke variable, man vil have med (og i hvilken rækkefølge).

Tilføjelsen `run;` i sidste linie er strengt taget heller ikke nødvendig, men kan stærkt anbefales efter hvert procedurekald, (og hvert `data-step`) da den såkaldte log-fil (indeholdende oplysninger om hvordan kørslen er forløbet, fejlmeddelelser mv.) herved bliver væsentlig lettere at læse.

7.2 Oversigt over datasæt, `proc contents`

Hvis man gerne vil have en summarisk oversigt over variabel navne, antal observationer mv. i et SAS-datasæt kan man skrive:

```
proc contents data = sasuser.bp ;
run;
```

7.3 Sortering, `proc sort`

```
proc sort data=a1;
  by fedme;
run;
```

Herved sorteres datasættet `a1` efter `fedme`. Hvis man i stedet ønsker at bevare datasættet `a1` og putte den sorterede version over i `a2`, skal man skrive:

```
proc sort data=a1 out=a2;
  by fedme;
run;
```

7.4 Tabellering, `proc freq`

Benyttes til at tabellere variable.

```
proc freq data=a1;
  tables sex;
run;
```

Bør kun bruges til diskrete variable med forholdsvis få værdier. Kontinuerte variable som kan antage mange forskellige værdier kan give meget store mængder af output.

Kan også benyttes til krydstabellering:

```
proc freq data=a1;
  tables sex * agr / norow nocol nopercnt ;
run;
```

7.5 Udregning af basale stikprøvestørrelser, `proc means` og `proc univariate`

For at se hvad denne procedure foretager sig, er det lettest at forsøge sig frem. Prøv f.eks. at skrive:

```
proc sort data=a1;
  by sex;
run ;
```

```
proc means data=a1;
  by sex;
run;
```

eller:

```
proc means data=a1;
  class sex;
run;
```

Bemærk, at man ved at skrive som ovenfor `by sex`; får udført den angivne procedure for hvert køn for sig (konstruktionen virker for stort set alle SAS-procedurer, men man *er nødt til at sortere observationerne først*, hvis de ikke allerede står i den rigtige rækkefølge).

Ønsker man uddybende oplysninger om en variabels fordeling, kan man også benytte `proc univariate`, f.eks. således:

```
proc univariate data = a plot normal ;
  by sex ;
  var fedme ;
run ;
```

Dette vil give os uddybende viden om fordelingen af `fedme` i vores stikprøve, opdelt efter køn. De to *options* `plot` og `normal` frembringer hhv. et box-plot og et test for normalitet.

8 Sammenklistring af datasæt

Det er kun nødvendigt at læse dette afsnit hvis man skal lave analyser med populations-baserede referencerater (SMR-analyser).

8.1 Datasæt i forlængelse af hinanden

To datasæt `a` og `b` kan lægges i forlængelse af hinanden til et nyt, `c` ved at skrive:

```
data c ;
  set a b ;
run ;
```

8.2 Datasæt ved siden af hinanden

To datasæt `a` og `b` kan lægges ved siden af hinanden til et nyt, `c`, ved at skrive:

```
data c ;
  merge a b ;
run ;
```

Herved bliver observation nr. 1 i datasæt `a` lagt ved siden af 1. observation i datasæt `b`. Hvis de findes variable med samme navn i både `a` and `b` vil disse variable få deres værdi fra `b` (nemlig det datatsæt der er nævnt sidst).

8.3 Datasæt ved siden af hinanden — `matched`

I kohorte-analyser er man interesseret i at tilordne observationer af risikotid en tilhørende referencerate.

F.eks. vil man for et stykke follow-up tid i aldersklassen 45–49 år i perioden 1953–58 gerne have tilordnet den tilsvarende mortalitetsrate for den danske befolkning.

Til den ende kræves at kohorte datasættet indeholder variable, f.eks. `ald` og `per`, med koder for aldersklasse hhv. periode. I ovennævnte eksempel kunne de f.eks. have værdierne 45 hhv. 53. Endvidere kræves at der foreligger et datasæt med reference-rater fra befolkningen, hvor alder og periode har samme navn og er kodet på samme måde.

Hvis man yderligere sørger for at datasættene er sorteret på samme måde efter alder og periode kan man sætte datasættene sammen:

```
proc sort data = koh ;
  by ald per ;
run ;

proc sort data = rater ;
  by ald per ;
run ;

data sammen ;
  merge koh rater ;
  by ald per ;
run ;
```

Denne operation bevirker at selv om der er flere forekomster med samme værdier af `ald` og `per` i `koh` og kun én i `rater` vil værdien af de øvrige variable i `rater` blive tilordnet alle observationerne i det resulterende datasæt `sammen`.

Man vil ofte opleve at ratefilen indeholder observationer (kombinationer af `ald` og `per`), som ikke forekommer i kohorte filen. Sådanne observationer vil være i det resulterende datasæt med alle variable fra kohortedsættet sat til missing. Man kan udelukke disse fra det resulterende datasæt ved at lave en temporær variabel, f.eks. `ok`, som er 1 hvis `koh` bidrager med data og 0 ellers, og så bruge den til at selekttere observationer med `ok=1`:

```
data sammen ;
  merge koh (in = ok) rater ;
  by ald per ;
  if ok ;
run ;
```

9 Datoer i SAS

I epidemiologiske follow-up studier er det nødvendigt at kunne håndtere datoer og tidsintervaller. SAS har implementeret faciliteter til at regne med datoer, som er anvendelige i epidemiologiske studier. Det følgende gennemgår de væsentligste ting omkring dato-repræsentation i SAS.

SAS repræsenterer alle datoer internt som antal dage siden d. 1.1.1960. Datoen 18. februar 1952 er således internt repræsenteret som -2874 , og d. 17 juli 1987 som 10059.

Det har den fordel at man blot kan subtrahere datoer for at få antallet af dage imellem dem. SAS har ingen problemer med år 2000.

Det ville naturligvis være tosset hvis man ikke kunne få SAS til at læse og skrive datoer som vi er vant til det, så det er der adskillige faciliteter til; nogen af dem omtales nedenfor.

9.1 Indlæsning af datoer

Hvis man skal indlæse datoer kan man f.eks. skrive dem i en fil, f.eks. `c:\dir\dat.fil`, på formen:

```
14/07/1952
23/04/1998
08/10/1980
```

og indlæse dem med:

```
data a ;
  infile 'c:\dir\dat.fil' ;
  input inddat ddmmyy10. ;
run ;
```

Variablen `inddat` vil få værdierne hhv. -2727 , 13997 og 7586. Det der står efter variabelnavnet er et s.k. *informat*, dvs. en specifikation af hvordan datoen skal opfattes. Nedenfor er en lille tabel over eksempler på hvordan d. 14. juli 1952 kan læses med SAS, og de dertil hørende informater:

14/07/52	ddmmyy8.
14 jul 1952	date11.
071452	mmddy6.
52-07-14	yymmdd8.

Det sidste tal i *informat*-navnet angiver hvor mange felter datoen maksimalt må fylde.

Hvis man ikke angiver århundredet vil årstal på 20 eller derover blive henført til 1900 tallet og årstal under 20 til 2000-tallet. Dette skillepunkt kan reguleres med:

```
options yearcutoff=1950
```

Hvis man vil vide hvad den aktuelle værdi er skal man skrive:

```
proc options option = yearcutoff ;
run ;
```

9.2 Udskrivning af datoer.

Der er selvfølgelig ikke meget grin ved at få skrevet den interne talrepræsentation ud, så SAS kan naturligvis også skrive datoer ud på mange måder. Det bruger man *formater* til. Formater er specifikationer af hvordan den underliggende talværdi skal skrives. Det gøres ved i `proc`-steppet at skrive f.eks.:

```
proc print data = a ;
  format inddat ddmmyy10. ;
run ;
```

som giver:

```
14/07/1952
23/04/1998
08/10/1980
```

mens:

```
proc print data = a ;
  format inddat weekdatx. ;
run ;
```

giver:

```
Monday, 14 July 1952
Thursday, 23 April 1998
Wednesday, 8 October 1980
```

Bemærk at såvel informater som formater ender på “.”.

9.3 Faste datoer.

Man har naturligvis også af og til brug for at kunne operere med en fast dato (“date constant”) i et program. SAS-repræsentationen af datoen d. 27. marts 1976 kan man i et SAS-program skrive som `'27mar76'd` — bemærk pingerne (') og det efterfølgende d. Hvis man f.eks. skal censurere alle ved en bestemt dato, kan man bruge et udtryk som:

```
uddat = min ( '27mar76'd, doddat )
```

Funktionen `min` returnerer den mindste af de ikke-missing argumenter, så hvis de personer der endnu er under observation d. 27. marts 1976 er missing for `doddat` får man den rette udgangsdato, nemlig dødsdatoen for de der er døde før d. 27. marts 1976, og censureringsdatoen 27. marts 1976 for alle andre (inklusive dem der er døde *efter* d. 27. marts 1976).

Obs: Man kan *ikke* indlæse datoer ved at skrive dem på formen `'14jul52'd` i en fil.

9.4 Regning med datoer.

Når man trækker to datoer fra hinanden får man antallet af dage imellem dem. Hvis man vil have antallet af år, bør man tage højde for skudår ved at skrive:

```
pyrs = ( uddat - inddat ) / 365.25 ;
```

og hvis man har brug for måneder:

```
pmdr = ( uddat - inddat ) / (365.25/12) ;
```

(Det er en dårlig ide at erstatte `365.25/12` med `30.4375`, selv om det er præcis det samme — når man senere skal læse sit eget program er det nemlig umiddelbart klart at `365.25/12` er det gennemsnitlige antal dage i en måned, mens tallet `30.4375` for de færreste har nogen umiddelbar mening).

I visse datamaterialer foreligger dato måned og år i tre forskellige variable. SAS-funktionen `mdy` klarer konverteringen:

```
dato = mdy(mnd, dag, aar)
```

Bemærk den amerikanske rækkefølge af argumenterne. Hvis det sidste argument er mindre end 100 antages året at være i 1900-tallet, ellers skal årstallet være mellem 1582¹ og 20000 for at give et ikke-missing resultat.

De tre tilsvarende omvendte funktioner findes også — de hedder meget naturligt `day`, `month` og `year`:

```
data a ;
  inddat = '14jul51'd ;
  dag = day ( inddat ) ;
  mnd = month ( inddat ) ;
  aar = year ( inddat ) ;
  output :
run ;

proc print data = a ;
run ;
```

giver:

OBS	INDDAT	DAG	MND	AAR
1	-3093	14	7	1951

Bemærk at funktionen `year` altid returnerer året inklusiv århundredet.

9.5 Opgaver i dato-behandling

1. Hvilke ugedage er du selv og dine familiemedlemmer født?
2. Hvor mange dage gamle er du og din familie tilsammen i dag?
3. Hvilken dato er du og dine familie-medlemmer tilsammen 100 år?
4. Fogh-regeringen tiltrådte d. 27. november 2001. Hvilken dato har den siddet i 100 dage?

¹I 1582 indførte pave Gregorius d. 13. den s.k. gregorianske kalender til erstatning for den gamle julianske (efter Julius Cæsar), så udregning af gregorianske datoer før 1582 giver ingen mening, eftersom de aldrig ville stemme overens med datoer angivet i kilder fra datiden. I den gregorianske kalender er de hele århundreder ikke skudår, undtagen de som kan deles med 400 — således var 1700, 1800 og 1900 ikke skudår, mens år 2000 var. I år 1700 indførtes den gregorianske kalender i Danmark — dagen efter d. 18. februar 1700 var d. 1. marts. Kilde: Den Store Danske Encyklopædi.

Part II

Follow-up data i SAS

Det følgende er i vid udstrækning taget fra en lignende introduktion til behandling af follow-up data med programpakken STATA, som David Clayton & Michael Hills har skrevet til brug på de europæiske sommerkurser i epidemiologi i Firenze.

Kohorte-studier er basalt set studier hvorfra det er muligt at estimere *rater*. I den forstand er sædvanlige kliniske follow-upstudier også kohorte studier. Men i modsætning til kliniske overlevelsesstudier er epidemiologiske follow-up studier karakteriseret ved:

- Lang follow-up tid for mange individer.
- Kovariater der afhænger af tiden.
- Forsinket indgang — dvs. personer indgår i kohortene på forskellige tidspunkter.
- Flere forskellige tidsskalaer kan være af interesse samtidigt, f.eks. alder, kalendertid, tiden siden eksponering ...
- Flere typer af udfald (“events”) er af interesse.

10 Repræsentation af follow-up data

Et kohorte studie er et hvor man kan opgøre *risikotid* og *antal hændelser*. For hvert individ skal der således nødvendigvis være en angivelse af risikotiden, dvs. normalt en angivelse af den periode personen er under risiko for at opleve en hændelse (død, cancerdiagnose, ...), og en angivelse af hvorvidt hændelsen er indtruffet.

Follow-up data indeholder derfor mindst tre variable:

1. Indgangsdato — **Entry** — datovariabel
2. Udgangsdato — **eXit** — datovariabel
3. Status ved udgang — **Fail** — indikator-variabel (0/1)

Disse tre variable er specifikke for en type af outcome — hver type outcome kræver sit eget sæt af IUS-variable (Ind,Ud,Status).

For et givet udfald repræsenteres hver person altså af indgangsdato, udgangsdato og personens status ved udgang (event ja/nej, Survival/Failure).

Ovenstående forudsætter at personerne er under observation i ét sammenhængende interval. Hvis der er tale om flere intervaller kan hver person repræsenteres af flere sæt IUS-observationer, en for hvert interval. Det er så højst ved udgangen af det sidste interval at en event kan forekomme. Rationalet bag at repræsentere personers follow-up med flere IUS-observationer kan f.eks. være at personen har skiftet ekspositionsstatus midt i sin follow-up tid; således får man to sæt IUS-observationer med forskellig tilhørende ekspositionsstatus.

10.1 Gentagne events

Ovennævnte situation forudsætter at en person forlader risiko populationen når en event er indtruffet. Hvis de events man studerer kan indtræffe flere gange (børnefødsler, influenza, skilsmisse), må det defineres hvornår en person der har oplevet en event indtræder i risiko igen, og for hver genindtræden (f.eks. 10 mdr. efter børnefødsel, 2 uger efter influenza, dagen efter giftermål) vil personen så bidrage med en ny IUS-observation.

Sådanne data kan i princippet behandles på fuldstændigt samme måde som data for død og andre engangsforeteelser. I det omfang man mener at antallet af tidligere hændelser har indflydelse på fremtidige rater kan dette inddrages via baggrundsvariable der knyttes til hver IUS-observation.

11 Opdeling af risikotid

Når risikotiden i et kohortestudie skal opgøres efter kalendertid, alder, tid siden ansættelse osv. er det praktisk at udtrykke alle inddelingerne på samme tidsskala, nemlig kalendertidsskalaen.

Til dette kræves at man for hver inddeling definerer

- et nulpunkt,
- en skala (antal dage pr. enhed) og
- et sæt delepunkter.

Eksempelvis:

Inddeling i kalendertidsintervallerne 1.1.1943–31.12.1947, 1.1.1948–31.12.1952,... kan f.eks. gøres ved at sætte:

- Nulpunkt: 1.1.1900
- Skala: 365.25
- Delepunkter: 43, 48, 53,...

eller:

- Nulpunkt: 1.1.1943
- Skala: 365.25
- Delepunkter: 0, 5, 10,...

Inddeling i aldersklasserne 0, 1–4, 5–9, 10–14,... kan gøres ved at vælge :

- Nulpunkt: Personens fødselsdag.
- Skala: 365.25
- Delepunkter: 0, 1, 5, 10,...

Bemærk at nulpunktet er specifikt for hver person, og at intervallerne ikke behøver at være lige lange.

Inddeling efter tid siden ansættelse i intervallerne 0–1, 2–5, 5–10, 10+ år:

- Nulpunkt: Ansættelsesdato
- Skala: 365.25
- Delepunkter: 0,2,5,10,+∞

Inddeling efter kumuleret stråledosis i Gy:

- Nulpunkt: Startdato for eksponering.
- Skala: $(365.25/12) \times$ Antal Gy/måned
- Delepunkter: 0, 0.5, 1, 2, 3, 5, 10

Bemærk at ikke alene nulpunktet men også skalaen kan være individuel.

Når der er defineret en opdeling og dele-datoerne er fastlagt for hver person kan man opdele hver persons follow-up tid i et antal intervaller, svarende til hvert af intervallerne for f.eks. kalendertidsskalaen. For hvert af disse intervaller kan man så efter ønske foretage en videreinddeling efter alder og så fremdeles.

For at kunne opdele risikotiden efter et vilkårligt antal forskellige skalaer skal man altså blot bruge en mekanisme, som for given nulpunkt, skala og delepunkter tager et stykke follow-up tid og klipper det i stykker.

Dette er muligt med SAS-macroen %Lexis. En SAS-macro er et SAS-program som SAS selv laver om på inden det udføres. I dette tilfælde skal programmet modificeres med nulpunkt, skala og delepunkter.

Macroen hentes ved at skrive:

```
%inc 'Lexis.sas' ;
```

Herved læses filen `Lexis.sas` som indeholder macro-koden. Macroen kan så kaldes fra SAS ved at skrive `%Lexis` med passende argumenter.

Hvis ens follow-up-data f.eks. ligger i et SAS-datasæt `flup` med IUS-variablene `in_dat`, `ex_dat` og `event`, kan follow-up-tiden opdeles efter kalendertid med:

```
%Lexis ( data = flup,
         out = fu_kal,
         entry = in_dat,
         exit = ex_dat,
         fail = event,
         cuts = 0 to 100 by 5,
         scale = 365.25,
         origin = '01jan1900'd,
         left = per,
         risk = pyrs ) ;
```

Bemærk at argumenterne står i en parentes adskilt af kommaer. Rækkefølgen af argumenterne er ligegyldig, linjeskiftene ligeså.

Resultatet er SAS-datasættet `fu_kal`, som vil indeholde de samme variable som `flup`, samt de to nye variable `per` der indeholder venstre endepunkt for hvert af kalendertidsintervallerne, og `pyrs` der giver risikotiden på den definerede skala, dvs. i person-år.

`in_dat` vil blive omdefineret til at være startdatoen for hver af intervallerne, og `ex_dat` til at være slutdatoen. For alle intervaller undtagen det sidste sættes `event` til 0, og i det sidste sættes `event` til den værdi variabelen havde i `flup`.

Al follow-up tid før det første cut og efter det sidste cut ekskluderes fra output-filen.

Det vil i de fleste tilfælde være mere fornuftigt at bevare de oprindelige indgangs- og udgangsdatoer, og definere nye variable til endepunkterne i de enkelte follow-up intervaller. Hvis man sørger for at disse nye variable hedder hhv. `entry` og `exit` og status-indikatoren `fail`, behøver man ikke at angive disse i kaldet af `%Lexis`.

11.1 Eksempel

Følgende lille program indlæser follow-up data for tre fiktive personer, og opdeler deres risikotid efter alder og kalendertid.

```
%inc 'Lexis.sas' ;

title1 'Oprindeligt datasæt' ;
data flup ;
  input id bth_dat in_dat ex_dat event ;
  informat bth_dat in_dat ex_dat ddmmyy8. ; * Hvordan datoerne læses ;
  format bth_dat in_dat ex_dat entry exit ddmmyy10. ; * Hvordan datoerne skrives ;
  entry = in_dat ;
  exit = ex_dat ;
  fail = event ;
  cards ;
1 14/07/52 04/08/65 27/06/97 1
2 01/04/54 08/09/72 23/05/95 0
3 10/06/87 23/12/91 24/07/98 1
;
run ;

proc print data = flup ;
  var id bth_dat in_dat ex_dat event entry exit fail ;
run ;

title1 '1. opdeling: alder' ;
%Lexis (data = flup,
       out = ald,
       cuts = 0 to 100 by 5,
       scale = 365.25,
       origin = bth_dat,
       left = age ) ;

proc print data = ald ;
  var id bth_dat in_dat ex_dat event entry exit fail risk age ;
```

```
run ;

title1 '2. opdeling: periode' ;
%Lexis (data = ald,
        out = aldper,
        cuts = 43 to 98 by 5,
        scale = 365.25,
        origin = '01jan1900'd,
        left = per ) ;

proc print data = aldper ;
var id bth_dat in_dat ex_dat event entry exit fail risk age per ;
run ;
```

Bemærk at man successivt opdeler risikotiden for personerne i mindre og mindre bidder:

Oprindeligt datasæt 16:58 Sunday, April 28, 2002 1

Obs	id	bth_dat	in_dat	ex_dat	event	entry	exit	fail
1	1	14/07/1952	04/08/1965	27/06/1997	1	04/08/1965	27/06/1997	1
2	2	01/04/1954	08/09/1972	23/05/1995	0	08/09/1972	23/05/1995	0
3	3	10/06/1987	23/12/1991	24/07/1998	1	23/12/1991	24/07/1998	1

1. opdeling: alder 16:58 Sunday, April 28, 2002 2

Obs	id	bth_dat	in_dat	ex_dat	event	entry	exit	fail	risk	age
1	1	14/07/1952	04/08/1965	27/06/1997	1	03/08/1965	14/07/1967	0	1.94319	10
2	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1967	14/07/1972	0	5.00000	15
3	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1972	14/07/1977	0	5.00000	20
4	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1977	14/07/1982	0	5.00000	25
5	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1982	14/07/1987	0	5.00000	30
6	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1987	14/07/1992	0	5.00000	35
7	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1992	27/06/1997	1	4.95277	40
8	2	01/04/1954	08/09/1972	23/05/1995	0	08/09/1972	01/04/1974	0	1.56057	15
9	2	01/04/1954	08/09/1972	23/05/1995	0	01/04/1974	01/04/1979	0	5.00000	20
10	2	01/04/1954	08/09/1972	23/05/1995	0	01/04/1979	31/03/1984	0	5.00000	25
11	2	01/04/1954	08/09/1972	23/05/1995	0	31/03/1984	31/03/1989	0	5.00000	30
12	2	01/04/1954	08/09/1972	23/05/1995	0	31/03/1989	01/04/1994	0	5.00000	35
13	2	01/04/1954	08/09/1972	23/05/1995	0	01/04/1994	23/05/1995	0	1.14168	40
14	3	10/06/1987	23/12/1991	24/07/1998	1	23/12/1991	09/06/1992	0	0.46338	0
15	3	10/06/1987	23/12/1991	24/07/1998	1	09/06/1992	09/06/1997	0	5.00000	5
16	3	10/06/1987	23/12/1991	24/07/1998	1	09/06/1997	24/07/1998	1	1.12115	10

2. opdeling: periode 16:58 Sunday, April 28, 2002 3

Obs	id	bth_dat	in_dat	ex_dat	event	entry	exit	fail	risk	age	per
1	1	14/07/1952	04/08/1965	27/06/1997	1	04/08/1965	14/07/1967	0	1.94319	10	63
2	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1967	02/01/1968	0	0.46886	15	63
3	1	14/07/1952	04/08/1965	27/06/1997	1	02/01/1968	14/07/1972	0	4.53114	15	68
4	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1972	01/01/1973	0	0.46886	20	68
5	1	14/07/1952	04/08/1965	27/06/1997	1	01/01/1973	14/07/1977	0	4.53114	20	73
6	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1977	01/01/1978	0	0.46886	25	73
7	1	14/07/1952	04/08/1965	27/06/1997	1	01/01/1978	14/07/1982	0	4.53114	25	78
8	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1982	01/01/1983	0	0.46886	30	78
9	1	14/07/1952	04/08/1965	27/06/1997	1	01/01/1983	14/07/1987	0	4.53114	30	83
10	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1987	02/01/1988	0	0.46886	35	83
11	1	14/07/1952	04/08/1965	27/06/1997	1	02/01/1988	14/07/1992	0	4.53114	35	88
12	1	14/07/1952	04/08/1965	27/06/1997	1	14/07/1992	01/01/1993	0	0.46886	40	88
13	1	14/07/1952	04/08/1965	27/06/1997	1	01/01/1993	27/06/1997	1	4.48392	40	93
14	2	01/04/1954	08/09/1972	23/05/1995	0	08/09/1972	01/01/1973	0	0.31554	15	68
15	2	01/04/1954	08/09/1972	23/05/1995	0	01/01/1973	31/03/1974	0	1.24504	15	73
16	2	01/04/1954	08/09/1972	23/05/1995	0	31/03/1974	01/01/1978	0	3.75496	20	73
17	2	01/04/1954	08/09/1972	23/05/1995	0	01/01/1978	01/04/1979	0	1.24504	20	78
18	2	01/04/1954	08/09/1972	23/05/1995	0	01/04/1979	01/01/1983	0	3.75496	25	78
19	2	01/04/1954	08/09/1972	23/05/1995	0	01/01/1983	31/03/1984	0	1.24504	25	83
20	2	01/04/1954	08/09/1972	23/05/1995	0	31/03/1984	02/01/1988	0	3.75496	30	83
21	2	01/04/1954	08/09/1972	23/05/1995	0	02/01/1988	31/03/1989	0	1.24504	30	88
22	2	01/04/1954	08/09/1972	23/05/1995	0	31/03/1989	01/01/1993	0	3.75496	35	88
23	2	01/04/1954	08/09/1972	23/05/1995	0	01/01/1993	31/03/1994	0	1.24504	35	93
24	2	01/04/1954	08/09/1972	23/05/1995	0	31/03/1994	23/05/1995	0	1.14168	40	93
25	3	10/06/1987	23/12/1991	24/07/1998	1	23/12/1991	09/06/1992	0	0.46338	0	88
26	3	10/06/1987	23/12/1991	24/07/1998	1	09/06/1992	01/01/1993	0	0.56400	5	88
27	3	10/06/1987	23/12/1991	24/07/1998	1	01/01/1993	09/06/1997	0	4.43600	5	93

```
28 3 10/06/1987 23/12/1991 24/07/1998 1 09/06/1997 01/01/1998 0 0.56400 10 93
```

Bemærk at p.g.a. afrundingsfejl i forbindelse med at vi har sat et år til 365.25 dage, er sker det af og til at aldersklasserne skiller en dag før eller efter personens fødselsdag.

Hvis man ud fra et sådant datasæt ønsker et hurtigt overblik over raterne i forskellige grupper kan man bruge macroen %PYtab:

```
%inc 'PYtab.sas' ;

%PYtab ( data = aldper,
         class = age,
         event = status,
         risk = risk,
         scale = 1000,
         cilevel = 90 ) ;
```

der i dette ikke særligt interessante tilfælde giver:

```
Rates per 1000 and 90 % c.i. by age                                17:10 Sunday, April 28, 2002  4
```

age	fail	risk	rate	r_lo	r_hi
.	1	0.060626	16.495	3.1841	85.447
0	0	0.000463	0.000	0.0000	110.694
5	0	0.005000	0.000	0.0000	10.259
10	0	0.002507	0.000	0.0000	20.459
15	0	0.006561	0.000	0.0000	7.818
20	0	0.010000	0.000	0.0000	5.129
25	0	0.010000	0.000	0.0000	5.129
30	0	0.010000	0.000	0.0000	5.129
35	0	0.010000	0.000	0.0000	5.129
40	1	0.006094	164.084	31.6748	849.994

Den første linje hvor `age` er manglende (“.”), er det totale antal events og den totale risikotid og altså den totale rate med 90% konfidensinterval.

12 SMR-analyser (PYRS-udregninger)

Hvis man i et kohorte-studie ud over analyser af rater ønsker at sammenligne kohortens morbiditet eller mortalitet med en referencebefolkning som man har rater for må man sørge for at:

1. kohortens risikotid skal være opdelt i samme klasser m.h.t. køn, alder og kalendertid mv. som de foreliggende referencerater.
2. variablene i kohortedatasættet der klassificerer risikotid efter disse har samme navn som i datasættet med referencerater.
3. begge datasæt skal være sorteret på samme måde efter de klassificerende variable (køn, alder, kaldendertid).

SMR (Standardized Mortality Ratio / Standardized Morbidity Ratio) er defineret som forholdet mellem det observerede antal events i en kohorte og det forventede antal events hvis kohorten var underlagt de samme rater som reference populationen.

Det forventede antal udregnes som den gennemlevede risikotid multipliceret med referenceraterne, så den praktiske tilgang til udregningerne bliver følgende:

1. Det observerede antal events og den gennemlevede risikotid opgøres inden for de relevante intervaller med SAS-macroen %Lexis. Normalt vil man skulle kalde Lexis to gange, en gang til opdeling efter alder, og en gang til opdeling efter kalendertid.
2. Referenceraterne klistres på datasættet med risikotiden opdelt efter alder, kalendertid og køn.
3. Det forventede antal tilfælde i hvert enkelt interval udregnes ved multiplikation af risikotiden med referenceraterne.

4. SMR kan nu udregnes ved summering af observerede og forventede antal og division af disse, eventuelt opdelt i undergrupper.
5. Endvidere kan datasættet anvendes direkte til Poisson-regression af den relative risiko, såfremt statusindikatoren er kodet 1 for event og 0 for censureing, og logaritmen af de forventede antal bruges som offset-variabel.

Part III

Analyse af epidemiologiske data i SAS

13 proc freq

Som omtalt ovenfor under de generelle procedurer bruges `proc freq` til at tabellere data. Proceduren optæller hvor mange observationer i datasættet der for hver kombination af de variable man tabellerer efter.

Fra et case-kontrol-studie af malignt melanom, kan vi f.eks tabellere cases og kontroller efter øjenfarve ved:

```
10      proc freq data = melanom ;
11          table eyes * casecon ;
12      run ;
```

NOTE: There were 1400 observations read from the data set WORK.MELANOM.

NOTE: The PROCEDURE FREQ printed page 1.

NOTE: PROCEDURE FREQ used:

```
real time      0.09 seconds
cpu time       0.04 seconds
```

The FREQ Procedure

Table of eyes by casecon

eyes	casecon		Total
Frequency	0	1	
Percent			
Row Pct			
Col Pct			
0	123	64	187
	8.82	4.59	13.41
	65.78	34.22	
	13.33	13.59	
1	312	138	450
	22.38	9.90	32.28
	69.33	30.67	
	33.80	29.30	
2	488	269	757
	35.01	19.30	54.30
	64.46	35.54	
	52.87	57.11	
Total	923	471	1394
	66.21	33.79	100.00

Frequency Missing = 6

Bemærk at den variabel der nævnes først kommer *nedad*, den der nævnes sidst *henad*. Man får altid procenter på begge leder og totalprocenter. Det kan undgås ved at skrive:

```
table eyes * casecon / norow nocol nopercent ;
```

Proc freq kan udregne et sædvanligt χ^2 -test for uafhængighed samt de forventede værdier og de enkelte cellers bidrag til teststørrelsen ud:

```
14      proc freq data = melanom ;
15          table eyes * casecon / norow nocol nopercent
16                          chisq cellchisq expected ;
17      run ;
```

NOTE: The PROCEDURE FREQ printed page 2.

NOTE: PROCEDURE FREQ used:

```
real time      0.08 seconds
```

```

-----
      cpu time          0.04 seconds
-----
The FREQ Procedure

Table of eyes by casecon

eyes          casecon
Frequency     |
Expected      |
Cell Chi-Square|          0|          1| Total
-----
          0 |    123 |    64 |    187
            | 123.82 | 63.183 |
            | 0.0054 | 0.0106 |
-----
          1 |    312 |    138 |    450
            | 297.96 | 152.04 |
            | 0.662  | 1.2973 |
-----
          2 |    488 |    269 |    757
            | 501.23 | 255.77 |
            | 0.3491 | 0.6841 |
-----
Total          923      471      1394

```

Frequency Missing = 6

Statistics for Table of eyes by casecon

Statistic	DF	Value	Prob
Chi-Square	2	3.0084	0.2222
Likelihood Ratio Chi-Square	2	3.0317	0.2196
Mantel-Haenszel Chi-Square	1	0.9677	0.3253
Phi Coefficient		0.0465	
Contingency Coefficient		0.0464	
Cramer's V		0.0465	

Effective Sample Size = 1394

Frequency Missing = 6

Man ser at der ikke er nogen signifikant forskel på fordelingen af øjenfarver mellem cases og kontroller ($p=0.22$).

13.1 Tabellerede data

Af og til har man data som på forhånd er tabellerede, dvs. man ikke har data for de enkelte individer, men kun en tabel som f.eks:

	Age \geq 60		Age < 60	
	MI yes	MI no	MI yes	MI no
SBP \geq 140	9	115	20	596
SBP < 140	6	73	21	1171

Hvis man gerne vil tabellere disse tal og bruge `proc freq` til at regne på dem ville det være yderst upraktisk at skulle danne et datasæt med i dette tilfælde 2011 observationer.

Derfor tillader `proc freq` at man lader hver observation i datasættet repræsentere flere personer. Antallet af personer for en given kombination af variablene skal være i en variabel som man nævner i et `weight`-statement:

```

data KSTab55 ;
  input age $ sbp $ mi $ antal ;
  cards ;

```

```

ge60 ge140 ja 9
ge60 lt140 ja 6
ge60 ge140 nej 115
ge60 lt140 nej 73
lt60 ge140 ja 20
lt60 lt140 ja 21
lt60 ge140 nej 596
lt60 lt140 nej 1171
;
run ;

```

```

15      proc freq data = KSTab55 ;
16          tables sbp * mi / chisq measures ;
17          weight antal ;
18      run ;

```

NOTE: There were 8 observations read from the data set WORK.KSTAB55.

NOTE: The PROCEDURE FREQ printed page 1.

NOTE: PROCEDURE FREQ used:

```

real time      0.29 seconds
cpu time       0.05 seconds

```

The FREQ Procedure

Table of sbp by mi

sbp	mi		Total
	ja	nej	
ge140	29	711	740
	1.44	35.36	36.80
	3.92	96.08	
	51.79	36.37	
lt140	27	1244	1271
	1.34	61.86	63.20
	2.12	97.88	
	48.21	63.63	
Total	56	1955	2011
	2.78	97.22	100.00

Statistics for Table of sbp by mi

Statistic	DF	Value	Prob
Chi-Square	1	5.5641	0.0183
Likelihood Ratio Chi-Square	1	5.3556	0.0207
Continuity Adj. Chi-Square	1	4.9209	0.0265
Mantel-Haenszel Chi-Square	1	5.5613	0.0184

---<slettet>---

Estimates of the Relative Risk (Row1/Row2)

Type of Study	Value	95% Confidence Limits	
Case-Control (Odds Ratio)	1.8793	1.1037	3.1997
Cohort (Col1 Risk)	1.8448	1.1009	3.0914
Cohort (Col2 Risk)	0.9817	0.9655	0.9981

Sample Size = 2011

Variablen antal indeholder antallet personer for hver kombination af de øvrige variable, og ved `weight antal`-angivelsen vil SAS ikke blot optælle antallet af observationer i datasættet (som i denne sammenhæng ville være 2 for alle celler), men summen af `antal` for de pågældende observationer i datasættet.

Når man beder om `measures` får man udregnet det som SAS kalder relative risk, som for case-kontrolstudier er odds-ratio. Man kan let verificere at $(29 \times 1244) / (27 \times 711) = 1.879$, og at konfidensintervallet er udregnet som $1.879 \times \exp(1.96 \times \sqrt{1/29 + 1/27 + 1/711 + 1/1244})$. Option `relrisk` kan også bruges.

I dette tilfælde er odds-ratio for myocardieinfarkt (MI) mellem personer med systolisk blodtryk over hhv. under 140 mmHg altså 1.88 med et 95% c.i. (1.10–3.20).

13.2 Stratificeret analyse

En analyse af melanomstudiet efter case-kontrol-status mod forekomsten af naevi giver:

```
data melanom ;
  infile 'melanom.txt' firstobs=2 ;
  input casecon sex brevald agr hudfarve hair eyes fregner akutrea kronrea
         nvsmall nvlarge nvtot ant15 ;
  naevus = ( nvtot > 0 ) + nvtot - nvtot ;
  * Trick for at få missing for naevus hvis nvtot er missing ;
run ;

20      proc freq data = melanom ;
21      table naevus * casecon / norow nocol nopercnt
22      chisq measures ;
23      run ;
NOTE: There were 1400 observations read from the data set WORK.MELANOM.
NOTE: The PROCEDURE FREQ printed page 3.
NOTE: PROCEDURE FREQ used:
      real time          0.16 seconds
      cpu time           0.04 seconds
```

The FREQ Procedure

Table of naevus by casecon

naevus	casecon		Total
Frequency	0	1	
0	635	231	866
1	286	241	527
Total	921	472	1393

Frequency Missing = 7

Statistics for Table of naevus by casecon

Statistic	DF	Value	Prob
Chi-Square	1	53.1068	<.0001

--<slettet>---

Estimates of the Relative Risk (Row1/Row2)

Type of Study	Value	95% Confidence Limits	
Case-Control (Odds Ratio)	2.3164	1.8439	2.9100
Cohort (Col1 Risk)	1.3511	1.2372	1.4755
Cohort (Col2 Risk)	0.5833	0.5049	0.6739

Effective Sample Size = 1393

Frequency Missing = 7

Her får vi estimeret odds-ratio associeret med tilstedeværelsen af naevi til 2.31, med et 95% c.i. på (1.84–2.91).

Nu er melanom-studiet imidlertid aldersstratificeret, så hvis man ville udregne en odds-ratio for malignt melanom associeret med tilstedeværelsen af naevi bør man lave en stratificeret analyse, dvs. man skal opdele efter alder, se på odds-ratio i hvert aldersstratum, se efter om de ser ens ud, og endelig estimere en fælles odds-ratio.

Man kan stratificere efter en (eller flere) variable i `proc freq` ved at stille disse *foran* i tabelspecifikationen samt specificere `cmh` efter skråstregen. `cmh` refererer til Cochran-Mantel-Haenszel-testet som er det stratificerede test for om odds-ratio er 1. Samtidig får man Breslow-Day-testet for hypotesen om at odds-ratios er ens over strata.

Output bliver temmelig voluminøst, idet man får en separat analyse for hvert aldersstratum, plus en samlet analyse til sidst. Nedenfor er passende uddrag:

```

25      proc freq data = melanom ;
26          table agr * naevus * casecon / norow nocol nopercnt
27                          cmh measures ;
28      run ;
NOTE: There were 1400 observations read from the data set WORK.MELANOM.
NOTE: The PROCEDURE FREQ printed pages 4-9.
NOTE: PROCEDURE FREQ used:
      real time          0.23 seconds
      cpu time           0.09 seconds
The FREQ Procedure

```

Table 1 of naevus by casecon
Controlling for agr=20

naevus	casecon		Total
Frequency	0	1	
0	25	12	37
1	15	9	24
Total	40	21	61

Estimates of the Relative Risk (Row1/Row2)			
Type of Study	Value	95% Confidence Limits	
Case-Control (Odds Ratio)	1.2500	0.4264	3.6643
Cohort (Col1 Risk)	1.0811	0.7379	1.5839
Cohort (Col2 Risk)	0.8649	0.4316	1.7330

Table 2 of naevus by casecon
Controlling for agr=30

naevus	casecon		Total
Frequency	0	1	
0	85	29	114
1	51	36	87
Total	136	65	201

Frequency Missing = 1

Estimates of the Relative Risk (Row1/Row2)			
Type of Study	Value	95% Confidence Limits	
Case-Control (Odds Ratio)	2.0690	1.1358	3.7689
Cohort (Col1 Risk)	1.2719	1.0346	1.5638
Cohort (Col2 Risk)	0.6148	0.4114	0.9186

Table 3 of naevus by casecon
Controlling for agr=40

naevus	casecon		
Frequency	0	1	Total
0	160	65	225
1	74	48	122
Total	234	113	347

Estimates of the Relative Risk (Row1/Row2)			
Type of Study	Value	95% Confidence Limits	
Case-Control (Odds Ratio)	1.5967	1.0042	2.5387
Cohort (Col1 Risk)	1.1724	0.9936	1.3833
Cohort (Col2 Risk)	0.7343	0.5434	0.9921

Sample Size = 347

---<tabel 4-7 slettet>---

Summary Statistics for naevus by casecon
Controlling for agr

Cochran-Mantel-Haenszel Statistics (Based on Table Scores)				
Statistic	Alternative Hypothesis	DF	Value	Prob
1	Nonzero Correlation	1	52.9943	<.0001
2	Row Mean Scores Differ	1	52.9943	<.0001
3	General Association	1	52.9943	<.0001

Estimates of the Common Relative Risk (Row1/Row2)				
Type of Study	Method	Value	95% Confidence Limits	
Case-Control	Mantel-Haenszel	2.3149	1.8426	2.9081
(Odds Ratio)	Logit	2.3191	1.8433	2.9178
Cohort	Mantel-Haenszel	1.3521	1.2379	1.4770
(Col1 Risk)	Logit	1.3327	1.2207	1.4550
Cohort	Mantel-Haenszel	0.5824	0.5038	0.6733
(Col2 Risk)	Logit	0.5806	0.5020	0.6716

Breslow-Day Test for Homogeneity of the Odds Ratios

Chi-Square	6.9098
DF	6
Pr > ChiSq	0.3293

Effective Sample Size = 1393
Frequency Missing = 7

Her får vi estimeret odds-ratio associeret med tilstedeværelsen af naevi til 2.31, med et 95% c.i. på (1.84–2.91), præcis det samme som i analysen uden stratifikation. Der er altså tydeligvis ikke nogen confounding, hvilket skyldes at der ikke er nogen sammenhæng mellem naevus-forekomst (+/–) og alder.

Både Mantel-Haenszel og Logit estimatoren for den fælles odds-ratio udregnes under den antagelse at odds-ratio i alle strata kan antages at være den samme. Breslow-Day testet for homogenitet af odds-ratios tester netop denne hypotese. I dette tilfælde er der tydeligvis ens odds-ratios i alle aldersklasser.

14 proc genmod

Denne SAS-procedure kan anvendes såvel til logistisk regression som til Poisson-regression.

14.1 Analyse af kohortestudier

På s. 226 i Clayton & Hills er vist hvordan data ser ud som “frequency records”. Dette datasæt kan indlæses i SAS og analyseres med proc genmod med følgende program:

```

data ihd ;
  input eksp alder pyrs cases ;
  lpyrs = log( pyrs );
cards;
0 2 311.9 2
0 1 878.1 12
0 0 667.5 14
1 2 607.9 4
1 1 1272.1 5
1 0 888.9 8
;
run;

16      proc genmod data = ihd ;
17      class alder eksp ;
18      model cases = alder eksp / dist = poisson
19      offset = lpyrs
20      type3 ;
21      run;
NOTE: Algorithm converged.
NOTE: The scale parameter was held fixed.
NOTE: The PROCEDURE GENMOD printed page 1.
NOTE: PROCEDURE GENMOD used:
      real time          0.43 seconds
      cpu time           0.09 seconds

```

Bemærk at man for at få risikotiden ind i modellen skal bruge logaritmen af denne (og log er den naturlige logaritme), som s.k. *offset*-variabel, angivet med *offset*-option i *proc genmod*.

Her opfattes *cases*, antallet af events (*D*), som Poisson-fordelt (*dist=poisson*) og logaritmen af raten (*link=log*) afhænger additivt af *alder* og *eksp*; *class*-statementet får SAS til at generere de relevante dummy-variable for hvert af niveauerne af *alder* og *eksp*.

The GENMOD Procedure

Model Information

Data Set	WORK.IHD
Distribution	Poisson
Link Function	Log
Dependent Variable	cases
Offset Variable	lpyrs
Observations Used	6

Class Level Information

Class	Levels	Values
alder	3	0 1 2
eksp	2	0 1

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	2	1.6727	0.8364
Scaled Deviance	2	1.6727	0.8364
Pearson Chi-Square	2	1.6516	0.8258
Scaled Pearson X2	2	1.6516	0.8258
Log Likelihood		52.5435	

Algorithm converged.

Analysis Of Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald	95% Confidence Limits	Chi-Square	Pr > ChiSq
Intercept	1	-5.4177	0.4421	-6.2841	-4.5513	150.20	<.0001
alder	0 1	0.6920	0.4614	-0.2123	1.5964	2.25	0.1337

alder	1	1	0.1290	0.4754	-0.8027	1.0607	0.07	0.7861
alder	2	0	0.0000	0.0000	0.0000	0.0000	.	.
eksp	0	1	0.8697	0.3080	0.2659	1.4734	7.97	0.0048
eksp	1	0	0.0000	0.0000	0.0000	0.0000	.	.
Scale		0	1.0000	0.0000	1.0000	1.0000		

NOTE: The scale parameter was held fixed.

LR Statistics For Type 3 Analysis

Source	DF	Chi-Square	Pr > ChiSq
alder	2	4.02	0.1342
eksp	1	8.30	0.0040

Estimatet for corner-parameteren står ud for *Intercept*, og repræsenterer $\log(\text{rate})$ for *sidste* niveau af *alder* og *eksp*. Hvad det er for et kan ses i *Estimate*-søjlen hvor der står 0, altså *eksp*=1 og *alder*=2. For tekst variable vil SAS sætte niveauerne af en *class*-variabel i alfabetisk orden. Hvis variabelen er defineret som numerisk kommer niveauerne i numerisk orden.

Bemærk at *proc genmod* udregner deviance pr. default, og at der udregnes 95% konfidensintervaller for $\log(RR)$. Endelige har *type3* bevirket at der er lavet likelihood-ratio-tests for fjernelse af hver af de to klasse-variable.

Individuelle records

På s. 229 i Clayton & Hills' bog bemærkes det at den samlede likelihood for alle bidrag fra hver af follow-up intervallerne fra alle personer i studiet er det samme som fra tabellerede data. Naturligvis under forudsætning af at de forklarende variable kun antager diskrete værdier som kan tabelleres efter.

I praksis betyder det at det for kohorte-studier ikke er nødvendigt at tabellere %lexis-opdelte data inden analyse; man skal blot analysere sine 0/1 data (0 fra intervaller uden event, 1 fra intervaller med) som om de var Poisson-fordelte, og anvende logaritmen af follow-up-tiden i hvert interval som offset-variabel. Den type af analyse vil typisk have et antal observationer i datasættet som er flere gange større end antallet af individer i kohorten.

For sådanne data er den absolutte værdi af deviance dog uden mening, mens forskelle mellem deviance for forskellige modeller stadig vil have mening.

14.2 Analyse af case-kontrol studier

Case-control studier kan analyseres med logistiske regressionsmodeller for binomial-data, dvs. man skal angive både tæller og nævner. Hvis man derfor indlæser data som antal cases og kontroller skal man udregne det totale antal inden man laver analysen.

Nedenfor er vist hvordan tallene i tabel 23.2, p. 230 i Clayton & Hills kan indlæses og analyseres.

```

data bvac ;
  input bcg alder cases controls ;
  total = cases + controls ;
cards;
1 7 1 7593
0 7 1 11719
1 6 11 7143
0 6 14 10184
1 5 28 5611
0 5 22 7561
1 4 16 2208
0 4 28 8117
1 3 20 2438
0 3 19 5588
1 2 36 4356
0 2 11 1625
1 1 47 5245
0 1 6 1234
;
run ;

24      proc genmod data = bvac ;
25          class alder bcg ;
26          model cases/total = alder bcg / dist = bin

```

```

27                                     link = logit ;
28      run;
NOTE: Algorithm converged.
NOTE: The scale parameter was held fixed.
NOTE: The PROCEDURE GENMOD printed page 1.
NOTE: PROCEDURE GENMOD used:
      real time          0.41 seconds
      cpu time           0.08 seconds

```

Bemærk at man ved logistisk regression skal angive respons-variablen som en brøk. Dette er kun en syntaks-mæssig konvention og venstresiden i model statementet kan *ikke* erstattes af en variabel hvor man har udregnet andelen af cases.

The GENMOD Procedure

```

      Model Information
Data Set          WORK.BVAC
Distribution       Binomial
Link Function     Logit
Response Variable (Events)    cases
Response Variable (Trials)    total
Observations Used          14
Number Of Events           260
Number Of Trials          80882

```

```

      Class Level Information
Class      Levels  Values
alder      7      1 2 3 4 5 6 7
bcg        2      0 1

```

```

      Criteria For Assessing Goodness Of Fit
Criterion      DF      Value      Value/DF
Deviance       6      3.6002      0.6000
Scaled Deviance 6      3.6002      0.6000
Pearson Chi-Square 6      3.6989      0.6165
Scaled Pearson X2 6      3.6989      0.6165
Log Likelihood          -1644.0206

```

Algorithm converged.

```

      Analysis Of Parameter Estimates
Parameter      DF      Estimate      Standard      Wald 95% Confidence      Chi-      Pr > ChiSq
                DF      Estimate      Error      Limits      Square
Intercept      1      -8.8800      0.7103      -10.2721      -7.4879      156.31      <.0001
alder          1      4.1576      0.7222      2.7422      5.5731      33.14      <.0001
alder          2      4.1556      0.7233      2.7379      5.5733      33.01      <.0001
alder          3      3.9002      0.7253      2.4786      5.3217      28.92      <.0001
alder          4      3.8241      0.7237      2.4057      5.2426      27.92      <.0001
alder          5      3.5831      0.7212      2.1695      4.9967      24.68      <.0001
alder          6      2.6235      0.7349      1.1831      4.0640      12.74      0.0004
alder          7      0.0000      0.0000      0.0000      0.0000      .          .
bcg            0      -0.5471      0.1409      -0.8232      -0.2709      15.07      0.0001
bcg            1      0.0000      0.0000      0.0000      0.0000      .          .
Scale          0      1.0000      0.0000      1.0000      1.0000      .          .

```

NOTE: The scale parameter was held fixed.

Individuelle records

I praktiske situationer hvor et case-kontrol-studie skal analyseres vil data foreligge som individuelle records, med et antal variable for hver person på hver linje, herunder en case-kontrol indikator — typisk kodet 1 for cases og 0 for kontroller.

Her kan man sige at hver linje repræsenterer en person, hvoraf der så er enten 1 eller 0 cases, så nævneren er altid 1. Det skulle så betyde det at man skal danne en variabel som antager værdien 1 for alle personer i studiet. Imidlertid er dette ikke nødvendigt når nævneren er 1; man kan godt tillade sig at udelade “/en” i eksemplet nedenfor, og skrive `model casecon = hudfarve`.

```

data melanom ;
  infile 'melanom.txt' firstobs=2 ;
  input casecon sex brevald agr hudfarve hair eyes fregner akutrea kronrea

```

```

      nvsmall nvlarge nvtot ant15 ;
run ;

proc genmod ;
  class hudfarve ;
  model casecon = hudfarve / dist = bin
                        link = logit ;
run ;

```

The GENMOD Procedure

Model Information

```

Data Set          WORK.MELANOM
Distribution       Binomial
Link Function     Logit
Dependent Variable casecon
Observations Used 1390
Missing Values    10

```

Class Level Information

```

Class      Levels  Values
hudfarve   3      0 1 2

```

Response Profile

Ordered Value	casecon	Total Frequency
1	0	919
2	1	471

PROC GENMOD is modeling the probability that casecon='0'. One way to change this to model the probability that casecon='1' is to specify the DESCENDING option in the PROC statement.

---<slettet>---

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	0.4864	0.0942	0.3017	0.6710	26.66	<.0001
hudfarve	0 1	0.5060	0.1575	0.1973	0.8148	10.32	0.0013
hudfarve	1 1	0.1691	0.1279	-0.0815	0.4198	1.75	0.1860
hudfarve	2 0	0.0000	0.0000	0.0000	0.0000	.	.
Scale	0	1.0000	0.0000	1.0000	1.0000	.	.

Bemærk at man med den sædvanlige kodning af responset 0/1 får estimeret den gale størrelse, nemlig sandsynligheden for at være en kontrol (givet inklusion i studiet). SAS er dog så venlig at give en anvisning på hvordan man får det gjort rigtigt.

```

36      proc genmod descending ;
37      class hudfarve ;
38      model casecon = hudfarve / dist = bin
39      link = logit ;
40      run ;
NOTE: PROC GENMOD is modeling the probability that casecon='1'.
NOTE: Algorithm converged.
NOTE: The scale parameter was held fixed.
NOTE: The PROCEDURE GENMOD printed page 11.
NOTE: PROCEDURE GENMOD used:
      real time          0.41 seconds
      cpu time           0.10 seconds

```

The GENMOD Procedure

Model Information

```

Data Set          WORK.MELANOM
Distribution       Binomial
Link Function     Logit
Dependent Variable casecon
Observations Used 1390
Missing Values    10

```

Class Level Information

```

Class      Levels  Values
hudfarve   3      0 1 2

```

Response Profile

```

Ordered      Total
Value casecon Frequency
  1      1      471
  2      0      919

```

PROC GENMOD is modeling the probability that casecon='1'.

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	1387	1769.3491	1.2757
Scaled Deviance	1387	1769.3491	1.2757
Pearson Chi-Square	1387	1390.0000	1.0022
Scaled Pearson X2	1387	1390.0000	1.0022
Log Likelihood		-884.6746	

Algorithm converged.

Analysis Of Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence		Chi-Square	Pr > ChiSq
				Limits			
Intercept	1	-0.4864	0.0942	-0.6710	-0.3017	26.66	<.0001
hudfarve 0	1	-0.5060	0.1575	-0.8148	-0.1973	10.32	0.0013
hudfarve 1	1	-0.1691	0.1279	-0.4198	0.0815	1.75	0.1860
hudfarve 2	0	0.0000	0.0000	0.0000	0.0000	.	.
Scale	0	1.0000	0.0000	1.0000	1.0000		

NOTE: The scale parameter was held fixed.

Man ser at den eneste forskel fra før er at estimaterne har skiftet fortegn. Det vil sige at hudfarve 0 og 1 (mørk hhv. medium) har lavere risiko end 2 (lys).

15 Udregning af odds-ratio, rate ratio og konfidensintervaller

I SAS findes et maskineri til at fiske diverse ting ud af analyser over i datasæt til videre processing, Output Delivery System, ODS.

```

30      proc genmod data = bvac ;
31          class alder bcg ;
32          model cases/total = alder bcg / dist = bin
33                          link = logit ;
34          ods output ParameterEstimates = pe ;
35      run;

```

NOTE: Algorithm converged.

NOTE: The scale parameter was held fixed.

NOTE: The data set WORK.PE has 11 observations and 9 variables.

NOTE: The PROCEDURE GENMOD printed page 2.

NOTE: PROCEDURE GENMOD used:

```

real time          0.32 seconds
cpu time           0.05 seconds

```

ods-statementet laver et SAS-datasæt, pe, der ligner den del af outputtet der står under "Analysis of parameter estimates", med bl.a. variablene Estimate, LowerCL og UpperCL.

I logistisk regression er Estimate jo $\log(\text{OR})$ så OR er $\exp(\text{Estimate})$, og konfidensintervallet for OR fås ved samme transformation:

```

37      data pe ;

```

```

38         set pe ;
39         or = exp( estimate ) ;
40         or_lo = exp( LowerCL ) ;
41         or_hi = exp( UpperCL ) ;
42         run ;
NOTE: There were 11 observations read from the data set WORK.PE.
NOTE: The data set WORK.PE has 11 observations and 12 variables.
NOTE: DATA statement used:
      real time           0.01 seconds
      cpu time            0.01 seconds
43
44         proc print  data = pe ;
45             var Parameter Level1 Estimate LowerCL UpperCL or or_lo or_hi ;
46         run ;
NOTE: There were 11 observations read from the data set WORK.PE.
NOTE: The PROCEDURE PRINT printed page 3.
NOTE: PROCEDURE PRINT used:
      real time           0.01 seconds
      cpu time            0.01 seconds

```

Obs	Parameter	Level1	Estimate	LowerCL	UpperCL	or	or_lo	or_hi
1	Intercept		-8.8800	-10.2721	-7.4879	0.0001	0.0000	0.001
2	alder	1	4.1576	2.7422	5.5731	63.9204	15.5211	263.243
3	alder	2	4.1556	2.7379	5.5733	63.7923	15.4548	263.314
4	alder	3	3.9002	2.4786	5.3217	49.4102	11.9251	204.725
5	alder	4	3.8241	2.4057	5.2426	45.7929	11.0861	189.155
6	alder	5	3.5831	2.1695	4.9967	35.9853	8.7543	147.921
7	alder	6	2.6235	1.1831	4.0640	13.7844	3.2645	58.205
8	alder	7	0.0000	0.0000	0.0000	1.0000	1.0000	1.000
9	bcg	0	-0.5471	-0.8232	-0.2709	0.5786	0.4390	0.763
10	bcg	1	0.0000	0.0000	0.0000	1.0000	1.0000	1.000
11	Scale		1.0000	1.0000	1.0000	2.7183	2.7183	2.718

En fuldstændig parallel til dette kan bruges ved udregning af rate ratio (relativ risiko) og tilhørende konfidensintervaller fra kohorte-studier.

15.1 estimate-statement i proc genmod

En mere fleksibel måde at få estimaterne ud fra modellen med konfidensintervaller er at bruge `estimate`, som også muliggør at man kan få andre reference-grupper end den man oprindeligt havde i modellen. Prisen er naturligvis lidt skrive-arbejde.

I modellen for BCG-data havde vi 7 alders grupper, og altså seks egentlige parametre, samt en referencegruppe som SAS har sat til 0. Med `estimate` kan man få `proc genmod` til at udregne vilkårlige forskelle, f.eks kan forskellene i log-OR mellem aldersgrupper 2 hhv. 3 og aldersgruppe 1 fås ved:

```

proc genmod data = bvac ;
  class alder bcg ;
  model cases/total = alder bcg / dist = bin
                        link = logit ;
  estimate "Agr 2 vs. 1" alder -1 1 0 0 0 0 0 ;
  estimate "Agr 3 vs. 1" alder -1 0 1 0 0 0 0 ;
run;

```

Det første efter “`estimate`” er blot en tekst så man selv kan huske hvad der er hvad. “`alder`” angiver at det alene er alderseffekterne man interesserer sig for. Da der er 7 aldersklasser skal der være 7 efterfølgende tal. Tallene “`-1 1 0 0 0 0 0`” angiver at man vil have $-1 \times 1.$ aldersparameter $+ 1 \times 2.$ aldersparameter $+ 0 \times 3.$ aldersparameter $+ \dots$, altså $4.1556 - 4.1576 = -0.0020$. Det er jo ret trivielt selv at regne ud, men den tilhørende spredning er ikke. Nedenfor kan man se hvordan outputtet fra `estimate` ser ud, og sammenligne med det output man (også) får for parameterestimaterne.

Analysis Of Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald	95% Confidence Limits	Chi-Square	Pr > ChiSq
Intercept	1	-8.8800	0.7103	-10.2721	-7.4879	156.31	<.0001
alder	1	4.1576	0.7222	2.7422	5.5731	33.14	<.0001
alder	2	4.1556	0.7233	2.7379	5.5733	33.01	<.0001

```
alder      3      1      3.9002      0.7253      2.4786      5.3217      28.92      <.0001
....
```

Contrast Estimate Results

Label	Estimate	Standard Error	Alpha	Confidence Limits	Chi-Square	Pr > ChiSq
Agr 2 vs. 1	-0.0020	0.2014	0.05	-0.3967 0.3927	0.00	0.9920
Agr 3 vs. 1	-0.2575	0.2210	0.05	-0.6906 0.1756	1.36	0.2439

Yderligere kan man få `proc genmod` til at udregne selve OR (hhv. RR) ved at tilføje “/ exp” til estimate-statementet. Hvis man f.eks vil sammenligne de resultater men får ved at udregne odds-ratios med aldersgruppe 7 som reference med de man får ved at bruge aldersgruppe 4 som reference kan man bruge:

```
49      proc genmod data = bvac ;
50          class alder bcg ;
51          model cases/total = alder bcg / dist = bin
52                      link = logit ;
55          estimate "Agr 1 vs. 7" alder 1 0 0 0 0 0 -1 / exp ;
56          estimate "Agr 2 vs. 7" alder 0 1 0 0 0 0 -1 / exp ;
57          estimate "Agr 3 vs. 7" alder 0 0 1 0 0 0 -1 / exp ;
58          estimate "Agr 4 vs. 7" alder 0 0 0 1 0 0 -1 / exp ;
59          estimate "Agr 5 vs. 7" alder 0 0 0 0 1 0 -1 / exp ;
60          estimate "Agr 6 vs. 7" alder 0 0 0 0 0 1 -1 / exp ;
61          estimate "Agr 1 vs. 4" alder 1 0 0 -1 0 0 0 / exp ;
62          estimate "Agr 2 vs. 4" alder 0 1 0 -1 0 0 0 / exp ;
63          estimate "Agr 3 vs. 4" alder 0 0 1 -1 0 0 0 / exp ;
64          estimate "Agr 5 vs. 4" alder 0 0 0 -1 1 0 0 / exp ;
65          estimate "Agr 6 vs. 4" alder 0 0 0 -1 0 1 0 / exp ;
66          estimate "Agr 7 vs. 4" alder 0 0 0 -1 0 0 1 / exp ;
67      run;
```

NOTE: Algorithm converged.

NOTE: The scale parameter was held fixed.

NOTE: The PROCEDURE GENMOD printed pages 4-5.

NOTE: PROCEDURE GENMOD used:

```
real time      0.07 seconds
cpu time       0.03 seconds
```

Contrast Estimate Results

Label	Estimate	Standard Error	Alpha	Confidence Limits	Chi-Square	Pr > ChiSq
Agr 1 vs. 7	4.1576	0.7222	0.05	2.7422 5.5731	33.14	<.0001
Exp(Agr 1 vs. 7)	63.9204	46.1618	0.05	15.5211 263.2429		
Agr 2 vs. 7	4.1556	0.7233	0.05	2.7379 5.5733	33.01	<.0001
Exp(Agr 2 vs. 7)	63.7923	46.1433	0.05	15.4548 263.3136		
Agr 3 vs. 7	3.9002	0.7253	0.05	2.4786 5.3217	28.92	<.0001
Exp(Agr 3 vs. 7)	49.4102	35.8360	0.05	11.9251 204.7253		
Agr 4 vs. 7	3.8241	0.7237	0.05	2.4057 5.2426	27.92	<.0001
Exp(Agr 4 vs. 7)	45.7929	33.1406	0.05	11.0861 189.1553		
Agr 5 vs. 7	3.5831	0.7212	0.05	2.1695 4.9967	24.68	<.0001
Exp(Agr 5 vs. 7)	35.9853	25.9534	0.05	8.7543 147.9209		
Agr 6 vs. 7	2.6235	0.7349	0.05	1.1831 4.0640	12.74	0.0004
Exp(Agr 6 vs. 7)	13.7844	10.1306	0.05	3.2645 58.2053		
Agr 1 vs. 4	0.3335	0.2193	0.05	-0.0963 0.7633	2.31	0.1283
Exp(Agr 1 vs. 4)	1.3959	0.3061	0.05	0.9082 2.1454		
Agr 2 vs. 4	0.3315	0.2220	0.05	-0.1036 0.7666	2.23	0.1354
Exp(Agr 2 vs. 4)	1.3931	0.3093	0.05	0.9016 2.1525		
Agr 3 vs. 4	0.0760	0.2210	0.05	-0.3571 0.5092	0.12	0.7308
Exp(Agr 3 vs. 4)	1.0790	0.2385	0.05	0.6997 1.6640		
Agr 5 vs. 4	-0.2410	0.2099	0.05	-0.6525 0.1705	1.32	0.2510
Exp(Agr 5 vs. 4)	0.7858	0.1650	0.05	0.5207 1.1858		
Agr 6 vs. 4	-1.2006	0.2528	0.05	-1.6961 -0.7050	22.55	<.0001
Exp(Agr 6 vs. 4)	0.3010	0.0761	0.05	0.1834 0.4941		
Agr 7 vs. 4	-3.8241	0.7237	0.05	-5.2426 -2.4057	27.92	<.0001
Exp(Agr 7 vs. 4)	0.0218	0.0158	0.05	0.0053 0.0902		

Her ser man bl.a. at spredningen på $\ln(\text{OR})$ er meget mindre når man bruger gruppe 4 som reference, idet der er mange flere cases og kontroller i denne end i aldersgruppe 7.

Man skal ikke bruge standard error for OR (eller RR), udelukkende estimatet og konfidensgrænserne, som er udregnet korrekt på log-skalaen og transformeret tilbage til OR-skalaen.

Det er ikke alle talkombinationer af parameter-estimerne det er tilladt at bede om, man kan kun få udregnet s.k. estimable funktioner. Populært sagt betyder det at summen af tallene skal være 0.

`Estimate` kan også bruges hvis der er kontinuerte variable i modellen. Så vil man skulle angive hvilket blodtryk, højde (eller hvad det nu er) man vil have ganget på koefficienten.