Analyse av generaliserte lineære modeller i statistikkpakken Splus

Ørnulf Borgan og Sven Ove Samuelsen

Desember 1993

1 Formål

Formålet med dette notatet er å gi en kort, uformell innføring i generaliserte lineære modeller (GLM) og å vise hvordan slike kan tilpasses i programsystemet Splus. Det forutsettes at leseren har elementære kunnskaper i Splus svarende til notatet "Starthjelp i S-PLUS".

2 Litteratur og dokumentasjon

Den sentrale boken om GLM er

> McCullagh & Nelder (1989). *Generalized linear models (2. edition).*
> Chapman & Hall.

En enklere innføring er

> Dobson (1983). *An introduction to statistical modelling.*
> Chapman & Hall.

En bok som blant annet beskriver i detalj hvordan GLM tilpasses i Splus er

> Chambers & Hastie (1992). *Statistical Models in S.*
> Wadsworth & Brooks/Cole.

En ytterligere nyttig bok om GLM, som legger stor vekt på anvendelser av statistikksystemet GLIM, er

> Aitkin, Anderson, Fransis & Hinde (1989). *Statistical modelling in GLIM.*
> Clarendon Press, Oxford.

3 Beskrivelse av tre eksempler

Nedenfor beskrives kort tre eksempler som vil bli benyttet til illustrasjon i dette notatet.

Eksempel 1 – Fødselsvekt og svangerskapets varighet

Dobson (1983, op.cit.) gir nedenstående tabell over sammenhengen mellom fødselsvekt og svangerskapets varighet for 24 babyer. (Tallene stammer ikke fra Norge. Norske babyer veier ca. 500 g mer.)

	Gutter		Jenter	
	Varighet (uker)	Fødselsvekt (g)	Varighet (uker)	Fødselsvekt (g)
	40	2968	40	3317
	38	2795	36	2729
	40	3163	40	2935
	35	2925	38	2754
	36	2625	42	3210
	37	2847	39	2817
	41	3292	40	3126
	40	3473	37	2539
	37	2628	36	2412
	38	3176	38	2991
	40	3421	39	2875
	38	2975	40	3231
Gjennomsnitt	38.33	3024.00	38.75	2911.33

En er interessert i å studere veksthastigheten pr. uke i slutten av svangerskapet, og om denne er forskjelig for de to kjønn.

Eksempel 2 – Dødelig giftdose for biller

Dobson (1983, op.cit.) gjengir data for en studie av effekten av CS_2 på dødeligheten av biller. I studien ble ca. 60 biller utsatt for hver av 8 ulike konsentrasjoner av CS_2, og antallet som døde ved hver av konsentrasjonene ble registrert. Resultatet ble som gitt i tabellen nedenfor.

Dose ($\log_{10} CS_2 \mathrm{mg\,l}^{-1}$)	Antall biller	Antall døde
1.6907	59	6
1.7242	60	13
1.7552	62	18
1.7842	56	28
1.8113	63	52
1.8369	59	53
1.8610	62	61
1.8839	60	60

En er interessert i å studere sammenhengen mellom dose og dødelighet for billene.

Eksempel 3 – Forekomst av lungekreft i 4 danske byer

Andersen (1977, *Scandinavian Journal of Statistics* **4**, 153–158) gjengir data fra en studie av lungekreft insidensen i fire danske byer i perioden 1968–71. Studien ble iverksatt fordi en hadde mistanke om en overhyppighet av lungekreft i industribyen Fredericia. Dataene fra studien er som gjengitt i tabellene nedenfor.

Tabell 1. *Observert antall lungekreft tilfeller mellom 1968 og 1971 i fire danske byer.*

Alder	By: Fredericia	Horsens	Kolding	Vejle	Totalt
40–54	11	13	4	5	33
55–59	11	6	8	7	32
60–64	11	15	7	10	43
65–69	10	10	11	14	45
70–74	11	12	9	8	40
> 75	10	2	12	7	31
Totalt	64	58	51	51	224

Tabell 2. *Antall innbyggere i de fire byene fordelt på aldersgrupper.*

Alder	By: Fredericia	Horsens	Kolding	Vejle	Totalt
40–54	3059	2879	3142	2520	11600
55–59	800	1083	1050	878	3811
60–64	710	923	895	839	3367
65–69	581	834	702	631	2748
70–74	509	634	535	539	2217
> 75	605	782	659	619	2665

4 Om generaliserte lineære modeller

De generaliserte lineære modeller består av en klasse av modeller som naturlig generaliserer de lineærnormale modeller. En presis beskrivelse er gitt i McCullagh & Nelder (1989, Kap. 2). Her vil vi nøye oss med en uformell introduksjon.

Vi tar utgangspunkt i den velkjente lineærnormale modellen. Denne er gitt ved at $Y_1, Y_2, \ldots, Y_n$ er uavhengige og normalt fordelte stokastiske variable med samme varians σ^2 og med

$$EY_i = \mu_i = \beta_0 x_{i0} + \beta_{i1} x_{i1} + \cdots + \beta_p x_{ip}\,. \tag{1}$$

Her er x_{ij}-ene gitte størrelser ofte kalt kovariater, forklaringsvariable eller uavhengige variable. Det er velkjent at x_{ij}-ene kan være kvantitative variable som i multippel regresjon eller indikatorvariable (0,1-variable) som i variansanalyse modeller.

Idéen med GLM er å generalisere den lineærnormale modellen både med hensyn til fordelingsformen og forventningsstrukturen. Vi betrakter følgende situasjon: $Y_1, Y_2, \ldots, Y_n$ er uavhengige stokastiske variable som alle er enten normalfordelt (med samme varians), gammafordelt (med samme skalaparameter), binomisk fordelt eller Poisson fordelt. Mer generelt kan Y_i-ene tilhøre en gitt fordeling i den eksponentielle fordelingsklassen. Vi lar $\mu_i = EY_i$ og antar at det fins en funksjon $g(\cdot)$ – kalt *linkfunksjonen* – som er slik at

$$g(EY_i) = g(\mu_i) = \beta_0 x_{i0} + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\,. \tag{2}$$

En generalisert lineær modell er således gitt ved følgende tre elementer:

(i) fordelingen

(ii) linkfunksjonen

(iii) det lineære uttrykket på høyre side av (2)

Den lineærnormale modellen svarer altså til at fordelingen er den normale med varians σ^2 og at linkfunksjonen er identitetsfunksjonen.

En annen interessant modell fås når Y_i-ene er binomisk (n_i, p_i) fordelte og linkfunksjonen er gitt ved

$$g(\mu_i) = \log\Big(\frac{\mu_i}{n_i - \mu_i}\Big) = \log\Big(\frac{p_i}{1 - p_i}\Big)\,. \tag{3}$$

Ved (2) fås da

$$p_i = \frac{e^{\beta_0 x_{i0} + \cdots + \beta_p x_{ip}}}{1 + e^{\beta_0 x_{i0} + \cdots + \beta_p x_{ip}}}\,, \tag{4}$$

som er modellen for logistisk regresjon.

Som en siste interessant modell nevnes her situasjonen hvor Y_i-ene er Poisson-fordelte med parametere μ_i og linkfunksjonen er

$$g(\mu_i) = \log(\mu_i). \tag{5}$$

Da fås ved (2) modellen

$$\mu_i = e^{\beta_0 x_{i0} + \cdots + \beta_p x_{ip}}, \tag{6}$$

som gir en log-lineær Poisson modell.

Vi vil konkretisere disse tre modell-typene ved eksemplene fra avsnitt 3.

Eksempel 1 – fortsatt

Vi lar Y_{ij}; for $i = 1, \ldots, 12$ og $j = 1, 2$; betegne fødselsvekten for barn nr. i av kjønn nr. j ($j = 1$ svarer til gutt og $j = 2$ til jente), og lar x_{ij}-ene være de tilsvarende varighetene avsvangerskapene. En rimelig første modell fås ved å anta at Y_{ij}-ene er uavhengige og normalfordelte med samme varians og med

$$\mu_{ij} = EY_{ij} = \alpha_j + \beta_j x_{ij} \,. \tag{7}$$

Dette er en modell hvor vi har separate regresjonslinjer for hvert kjønn. Vi er interessert i å undersøke om vekthastighetene β_1 og β_2 for de to kjønn er like. Hypotesen om lik veksthastighet testes som kjent ved

$$F = \frac{(SS_{H_0} - SS_{\mathrm{apr}})/1}{SS_{\mathrm{apr}}/20} \,, \tag{8}$$

hvor SS_{apr} og SS_{H_0} er residual kvadratsummen under modellen (7) og under modellen som fås når $\beta_1 = \beta_2 = \beta$. Under nullhypotesen er F testobservatoren F-fordelt med 1 og 20 frihetsgrader.

Eksempel 2 – fortsatt

Vi lar her Y_i angi antall biller som dør ut av n_i ved dose x_i; $i = 1, 2, \ldots, 8$. Det er rimelig å anta at Y_i-ene er binomisk fordelte (n_i, p_i). En mulig modell som forbinder dødssannsynlighetene p_i med dosene x_i er

$$p_i = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \,, \tag{9}$$

som er et spesialtilfelle av den logistiske regresjonsmodell (4).

Eksempel 3 – fortsatt

La her Y_{ij}; for $i = 1, 2, 3, 4$ og $j = 1, 2, \ldots, 6$; betegne antall lungekrefttilfeller i by nr. i (1 = Fredericia; 2 = Horsens; 3 = Kolding; 4 = Vejle) og aldersintervall nr. j og la n_{ij} angi de tilsvarende antall innbyggere i de relevante byer og aldersgrupper. Det er rimelig å anta at Y_{ij}-ene er Poisson fordelte med parametere proporsjonale med n_{ij}-ene. Spesielt vil vi betrakte en modell med multiplikativ effekt av by og alder, dvs. en modell av formen

$$\mu_{ij} = EY_{ij} = n_{ij} \varphi_i \theta_j \,.$$

Med $x_{ij0} = \log n_{ij}$, $\alpha_i = \log \varphi_i$ og $\beta_j = \log \theta_j$ kan denne skrives som

$$\mu_{ij} = e^{x_{ij0} + \alpha_i + \beta_j}, \tag{10}$$

dvs. av formen (6), men med $\beta_0 = 1$. Vi er interessert i å estimere parameterene i denne modellen samt i å teste hypotesen om at alle α_i-ene er like.

5 Eksempler på GLM-analyser i Splus

Vi vil i dette avsnittet vise hvordan Splus kan benyttes til å analysere våre tre eksempler. Dette vil illustrerer bruken av de mest sentrale Splus-kommandoene i forbindelse med GLM.

Eksempel 1 – fortsatt

Filen

```
~borgan/data/foedselsvekt
```

inneholder datene i Eksempel 1. For å kunne gjøre de påfølgende beregningene selv må denne først kopieres over til eget område. Filen ser ut som følger:

1	40	2968
1	38	2795
1	40	3163
1	35	2925
1	36	2625
1	37	2847
1	41	3292
1	40	3473
1	37	2628
1	38	3176
1	40	3421
1	38	2975
2	40	3317
2	36	2729
2	40	2935
2	38	2754
2	42	3210
2	39	2817
2	40	3126
2	37	2539
2	36	2412
2	38	2991
2	39	2875
2	40	3231

På filen er det én linje for hver baby. Videre angir første søyle babyens kjønn (1=gutt, 2=jente), andre søyle svangerskapets varighet (i uker) mens tredje søyle er fødselsvekten (i gram).

Vi leser filen inn i Splus ved kommandoen[1]

```
foedselsvekt<-matrix(scan(''foedselsvekt''),ncol=3,byrow=T)
```

Det er hensiktsmessig å gi navn til variablene som angir kjønn, svangerskapets varighet og fødselsvekt. Dette gjøres ved kommandoene[2]

[1] Filen `foedselsvekt` må være lagret under det directoriet Splus startes fra.

[2] Alternativt kan innlesingen og navngivingen gjøres under ett med kommandoen `foedsel<-read.table(''foedselsvekt'',col.names=c(''kjonn'',''varighet'',''vekt''))`. En får da generert en såkalt "dataframe". Variablene i denne vil enkeltvis ha navnene `foedsel$kjonn`, `foedsel$varighet` og `foedsel$vekt`.

```
kjonn<-foedselsvekt[,1]; varighet<-foedselsvekt[,2]; vekt<-foedselsvekt[,3]
```

For å kunne analysere dataene må Splus få nærmere beskjed om hva slags type informasjon de ulike datavektorene inneholder. Splus skiller her mellom kvalitative variable, eller faktorer, og kvantitative variable. For en faktor betyr ikke verdien i seg selv noe. Den er bare en kode for et bestemt nivå av faktoren, slik som 1 for gutt og 2 for jente for faktoren `kjonn`. For en kvantitativ variabel har derimot verdien direkte mening, slik som antall uker svangerskapet har vart og babyens fødselsvekt. For å gi Splus beskjed om at `kjonn` skal være en faktor gir vi kommandoen[3]

```
kjonn<-factor(kjonn)
```

Faktorer kan representeres på forskjellige måter i Splus, med terminologi fra variansanalysen som forskjellige kontraster. Ulike valg av kontraster svarer til ulike parameteriseringer av modellen. En type kontrast som benyttes ofte i GLM er såkalte indikator- eller behandlingskontraster. For disse sammenlignes nivå j av faktoren (svarende til behandling nr. j) med et referansenivå $j = 1$ (standardbehandlingen). Vi vil i dette notatet konsentrere oss om denne typen kontraster.

Kommandoen i Splus for GLM-analyser har mye struktur felles med kommandoen for analyse av vanlige linærnormale modeller. Da det innen varianseanalysen er vanlig å bruke andre kontraster, vil Splus ikke benytte indikator/behandlings-kontrastene med mindre vi gir eksplisit beskjed om dette. Hvis en har faktorer i modellen, og som her ønsker å benytte indikator/behandlings-kontrastene, må en alltid gi kommandoen

```
options(contrasts=''contr.treatment'')
```

før en tilpasser modellen[4].

Variable som ikke er definert som faktorer oppfatter Splus som kvantitative (såsant de ikke er "boolske", dvs tar verdiene T eller F). Vi trenger altså ikke å redefinere variablene `varighet` og `vekt`.

Vi kan nå tilpasse modellen foreslått i avsnitt 4, dvs. at fødselsvekten er normalfordelt og at det er forskjellige regresjonslinjer for gutter og jenter, ved kommandoen[5]

```
glm(vekt~kjonn+kjonn:varighet-1,family=gaussian())
```

Uttrykket

```
vekt~kjonn+kjonn:varighet-1
```

[3]Hvis vi leser inn dataen ved hjelp av `read.table` slik det ble angitt i forrige fotnote, må vi her gi kommandoen `foedsel$kjonn<-factor(foedsel$kjonn)`.

[4]Hvis vi ønsker den vanlige parameteriseringen fra variansanalysen hvor parametrene summerer til null må vi i stedet gi kommandoen `options(contrasts=''contr.sum'')`. Andre muligheter er å bruke ortogonale polynomer eller såkalte Helmert kontraster. Den siste av disse er default i Splus.

[5]Hvis vi leser inn dataen ved hjelp av `read.table`, må vi her gi kommandoen `glm(formula=vekt~ kjonn+kjonn:varighet-1,family=gaussian(),data=foedsel)`.

kalles for modellformelen. Denne angir for det første at `vekt` er responsvariablen Y. Modellformelen angir dessuten at fødselsvekten avhenger av kovariatene kjønn og svangerskapets varighet i henhold til modell (7). F.eks. oversetter Splus `kjonn` til leddet α_j i (7), mens `kjonn:uker` oversettes til leddet $\beta_j x_{ij}$. Leddet `-1` er tatt med for at Splus ikke skal ta med noe felles konstantledd i (7). Denne syntaksen svarer til den som benyttes i McCullagh & Nelder (1989, avsnitt 3.4) med unntak av at dot-operatoren '.' i Splus angis ved ':'.

At responsvariablen er normalfordelt angis ved `family=gaussian()`. For normalfordelte variable tillater Splus bare identitetsfunksjonen som link[6], derfor står det ingenting om linkfunksjon i kommandoen over. Strengt tatt kan vi også utelate å angi fordelingen når vi har normalfordelte responser. Her er er det faktisk nok å gi kommandoen

```
lm(vekt~kjonn+kjonn:varighet-1)
```

siden denne betyr at en vanlig lineærnormal modell skal tilpasses.

Som svar på `glm`-kommandoen over gir Splus oss resultatet

```
Call:
glm(formula = vekt ~ kjonn + kjonn:varighet - 1, family = gaussian())

Coefficients:
    kjonn1    kjonn2 kjonn1varighet kjonn2varighet
 -1268.672 -2141.667       111.9828          130.4

Degrees of Freedom: 24 Total; 20 Residual
Residual Deviance: 652424.5
```

Her svarer `kjonn1` til $\hat{\alpha}_1$, `kjonn2` til $\hat{\alpha}_2$, `kjonn1varighet` til $\hat{\beta}_1$ og `kjonn2varighet` til $\hat{\beta}_2$. De tilpassede regresjonslinjene er altså

$$-1269 + 112.0\,\mathrm{x}$$

for gutter og

$$-2142 + 130.4\,\mathrm{x}$$

for jenter.

Splus angir også "Residual Deviance" med tilhørende antall frihetsgrader. En definisjon av devians-begrepet er gitt i McCullagh & Nelder (1989, avsnitt 2.3). Her er det nok å vite at i den aktuelle situasjonen med normalfordelte data og identitetsfunksjonen som linkfunksjon, så er deviansen det samme som residual kvadratsummen. Frihetsgradstallet på 20 framkommer naturlig nok ved at vi her har 24 observasjoner og 4 estimerte parametere.

Det er ofte en god ide å lagre resulatet av en GLM-tilpasning i et såkalt objekt. Tilpasning over lagres i objektet `foedsel.res` ved å skrive

[6] Normalfordelte data med annen linkfunksjon kan tilpasses på andre måter i Splus, men vi unnlater å komme inn på nærmere detaljer her.

```
foedsel.res<-glm(vekt~kjonn+kjonn:varighet-1,family=gaussian())
```

Gir vi nå kommandoen

```
foedsel.res
```

får vi ut resultatene fra GLM-tilpasningen over. Mer informasjon om GLM tilpasningen får vi ved å skrive

```
summary(foedsel.res)
```

Denne informasjonen består bl.a. i følgende tabell over estimerte regresjonskoeffisienter med tilhørende standardfeil og t-verdier.

```
Coefficients:
                    Value Std. Error   t value
        kjonn1 -1268.6724 1114.63840 -1.138192
        kjonn2 -2141.6667 1163.60146 -1.840550
kjonn1varighet   111.9828   29.04570  3.855400
kjonn2varighet   130.4000   29.99826  4.346918

(Dispersion Parameter for Gaussian family taken to be 32621.23 )
```

Dispersjonsparameteren er i dette tilfellet det vanlige estimatet for σ^2.

Etter at modellen (7) er tilpasset kan vi nå enkelt også tilpasse modellen hvor $\beta_1 = \beta_2 = \beta$. Dette gjøres ved å gi kommandoen

```
glm(vekt~kjonn+varighet-1,family=gaussian())
```

Her har `kjonn` samme betydning som over, mens `varighet` oversettes til βx_{ij}. Vi får altså modellen

$$\mu_{ij} = EY_{ij} = \alpha_j + \beta x_{ij}\,.$$

Splus gir for denne modellen resultatet

```
Call:
glm(formula = vekt ~ kjonn + varighet - 1, family = gaussian())

Coefficients:
    kjonn1    kjonn2 varighet
 -1610.283 -1773.322 120.8943

Degrees of Freedom: 24 Total; 21 Residual
Residual Deviance: 658770.7
```

Her er `kjonn1` og `kjonn2` som over $\hat{\alpha}_1$ og $\hat{\alpha}_2$, mens `varighet` er $\hat{\beta}$.

Ut fra resultatet for de to modell-tilpasningene kan nå F-observatoren (8) beregnes til

$$F = \frac{658770.7 - 652424.5}{652424.5/20} = 0.19,$$

så hypotesen om lik veksthastighet for de to kjønn kan ikke forkastes.

Eksempel 2 – fortsatt

Dataene skrives først på filen `biller` på følgende måte

```
1.6907  59   6
1.7242  60  13
1.7552  62  18
1.7842  56  28
1.8113  63  52
1.8369  59  53
1.8610  62  61
1.8839  60  60
```

dvs. nøyaktig som gitt i tabellen i avsnitt 3. (Dataene fins forøvrig på filen `~borgan/data/biller`.) Dataene leses inn og gjøres klare for analyse i Splus på samme måte som i forrige eksempel, dvs. ved kommandoene

```
biller<-matrix(scan(''biller''),ncol=3,byrow=T)
dose<-biller[,1]; antall<-biller[,2]; dode<-biller[,3]
```

Det er ingen faktorer i modellen så vi slipper å definer slike og å angi kontraster. Splus-kommandoen som legger tilpasningen av modellen (9) inn i objektet `biller.res` blir nå

```
biller.res<-glm(cbind(dode,antall-dode)~dose,family=binomial(link=logit))
```

At antall døde biller er binomisk fordelt angis altså ved å skrive `family=binomial`, og at modellen krever logit-linkfunksjonen (3) gis ved argumentet `link=logit`. Modellformelen består som i Eksempel 1 av både spesifikasjon av responsvariablen og av å angi hvordan det lineære leddet (2) ser ut. Som respons for binomiske data må Splus vite både hvor mange som døde, og hvor mange som overlevde[7]. Dette angis altså ved `cbind(dode,antall-dode)`. Siden det bare er én kovariat i modellen spesifiseres det lineære leddet $\beta_0 + \beta_1 x_i$ kun ved `dose`. Konstantleddet β_0 blir automatisk med siden vi her ikke har angitt `-1` i modellformelen.

Hvis vi nå skriver

```
biller.res
```

vil Splus svare med

```
Call:
glm(formula = cbind(dode, antall - dode) ~ dose, family = binomial(link = logit
        ))

Coefficients:
 (Intercept)     dose
   -60.71733 34.27026

Degrees of Freedom: 8 Total; 6 Residual
Residual Deviance: 11.23223
```

[7]Hvis vi har individuelle responser slik at Y-ene bare kan være lik 0 eller 1 er det nok å angi disse.

Her svarer `Intercept` til $\hat{\beta}_0$ og `dose` til $\hat{\beta}_1$. Den tilpassede modellen for dødssannsynligheten $p(x)$ som funksjon av dose x blir altså

$$p(x) = \frac{e^{-60.72+34.27x}}{1+e^{-60.72+34.27x}}$$

Hvis vi gir kommandoen `summary(biller.res)` ser vi at standardfeilen for $\hat{\beta}_1$ er 2.91 hvilket gir en "t-verdi" (Wald test) på 11.78. Det er altså en sterkt signifikant sammenheng mellom dose og dødelighet.

Tilpasningen til modellen kan vi sjekke fra "Residual Deviance" som her svarer til sannsynlighetskvotetesten for testing av nullhypotesen om at p_i-ene kan gis ved (9). Verdien av denne på 11.23 sammenlignet med kji-kvadrat fordelingen med $8-2=6$ frihetsgrader gir en P-verdi på 8,1%. Så selv om modellen (9) ikke forkastes på 5% nivå er ikke tilpasningen særlig god.

For nærmere å studere tilpasningen til modellen kan vi studere de standardiserte residualene (eller Pearson residualene). Disse fås ved kommandoen[8]

```
residuals(biller.res,"pearson")
```

Resultatet blir

```
        1        2       3         4         5          6        7        8
 1.406698 1.100159 -1.1761 -1.612308 0.5941439 -0.1279519 1.089336 1.130217
```

Vi ser at residualene er positive for små og store verdier av dosen og i hovedsak negative for midlere verdier. Dette nokså systematiske mønsteret i residualene antyder også at modellen ikke gir en fullt ut tilfredsstillende beskrivelse av dataene.

Eksempel 3 – fortsatt

Filen

```
~borgan/data/lungekreft
```

inneholder datene i Eksempel 3. For å kunne gjøre de påfølgende beregningene selv må denne først kopieres over til eget område. Filen ser ut som følger:

[8]Splus gir oss mulighet for å beregne flere andre typer residualer. Av særlig interesse er de såkalte devians residualene (som er default); se McCullagh & Nelder (1989), Avsnitt 2.4.

1	1	11	3059
1	2	11	800
1	3	11	710
1	4	10	581
1	5	11	509
1	6	10	605
2	1	13	2879
2	2	6	1083
2	3	15	923
2	4	10	834
2	5	12	634
2	6	2	782
3	1	4	3142
3	2	8	1050
3	3	7	895
3	4	11	702
3	5	9	535
3	6	12	659
4	1	5	2520
4	2	7	878
4	3	10	839
4	4	14	631
4	5	8	539
4	6	7	619

På filen er by gitt i første kolonne, aldersintervall i andre, antall lungekrefttilfeller i tredje og antall innbyggere i fjerde kolonne.

Vi gir så Splus følgende kommandoer

```
lungekreft<-matrix(scan("lungekreft"),ncol=4,byrow=T)
by<-lungekreft[,1]; alder<-lungekreft[,2]; kreft<-lungekreft[,3]
befolkning<-lungekreft[,4]
by<-factor(by); alder<-factor(alder)
options(contrasts=''contr.treatment'')
```

Betydningen av disse er tilsvarende som i de to foregående eksemplene. Et spesielt trekk ved modellen (10) er forekomsten av konstantleddet x_{ij0}. Et slikt ledd kalles en "offset". For å angi dette beregner vi først en vektor `logbef` som inneholder logaritmen til antall innbyggere. Dette gjøres ved

```
logbef<-log(befolkning)
```

Modellen (10), som sier at antall krefttilfeller er poissonfordelt med log-linkfunksjonen (5) og at det lineære leddet er lik offset pluss faktorene alder og by, kan nå tilpasses ved

```
kreft.res1<-glm(kreft~offset(logbef)+alder+by-1,family=poisson(link=log))
kreft.res1
```

hvilket gir resultatene

```
Call:
glm(formula = kreft ~ offset(logbef) + alder + by - 1, family = poisson(link =
        log))

Coefficients:
    alder1    alder2    alder3    alder4    alder5    alder6      by2
 -5.632065 -4.531051 -4.113452 -3.864358 -3.775201 -4.212411 -0.33006
        by3        by4
 -0.3715462 -0.2723177

Degrees of Freedom: 24 Total; 15 Residual
Residual Deviance: 23.44748
```

Siden vi spesifiserte modellen som `alder+by-1` (og ikke `by+alder-1`) gir estimatene `alder1, ..., alder6`, (svarende til β_j-ene i (10)) direkte effekten av et individs alder på lungekreft insidensen, mens `by2, by3` og `by4` (svarende til α_2, α_3 og α_4 i (10)) måler endringer i lungekreft insidens for byene Horsens, Kolding og Vejle relativt til Fredericia (siden `by1` $= \alpha_1$ er satt lik null pr. definisjon). Ikke overraskende finner vi en tendens til stigende lungekreft insidens med økende alder. Videre ser vi at Horsens, Kolding og Vejle alle har en estimert lungekreft insidens som er lavere enn Fredericias.

"Residual Deviance" gir her sannsynlighetskvotetesten for testing av nullhypotesen om at Poisson-parameterene kan gis på formen (10). Verdien 23.45 av denne sammenlignet med kji-kvadrat fordelingen med $24 - 9 = 15$ frihetsgrader gir en P-verdi på 7,7%. Modellen forkastes altså ikke, selv om tilpasningen ikke er spesielt god.

Ved å benytte kommandoen `residuals(kreft.res1,''pearson'')` som beskrevet i forrige eksempel vil vi se at 3 av de 24 standardiserte residualene er over 2 i absoluttverdi, nemlig de svarende til 40-54 år og over 75 år i Horsens og over 75 år i Vejle. Dette illustrerer igjen at modellen ikke er helt ut tilfredsstillende, men vi vil ikke behandle dette problemet videre her.

Hvis vi aksepterer modellen (10), kan vi teste om lungekreft insidensen er signifikant forskjellig i de fire byene ved å gi Splus kommandoen

```
kreft.res2<-glm(kreft~offset(logbef)+alder-1,family=poisson(link=log))
kreft.res2
```

som betyr at vi tilpasser samme modell bare med faktoren `by` fjernet fra modell (10). Resultatet blir

```
Call:
glm(formula = kreft ~ offset(logbef) + alder - 1, family = poisson(link = log))

Coefficients:
    alder1    alder2    alder3    alder4    alder5    alder6
 -5.862253 -4.779911 -4.360577 -4.111966 -4.015031 -4.453972
```

```
Degrees of Freedom: 24 Total; 18 Residual
Residual Deviance: 28.30653
```

"Residual Deviance" gir her sannsynlighetskvotetesten for testing av nullhypotesen om at Poisson-parameterene kan gis som i (10) men uten faktoren `by`. Endringen i "Residual Deviance" blir dermed sannsynlighetskvotetesten for testing av nullhypotesen om at alle α_i-ene i (10) er null når (10) aksepteres som "apriori-modell". Denne endringen på 28.31–23.45=4.86 skal sammenlignes med fraktilene i kji-kvadrat fordelingen med 3 frihetsgrader. Dette gir en ikke-signifikant P-verdi på 18,2%. Vi har altså ikke påvist noen økt lungekreft insidens i Fredericia.

Appendiks – Dokumentasjon av noen Splus kommandoer

I dette appendikset gjengir vi dokumentasjonen til `glm`, `glm.objekt` og `family`. Dokumentasjonen er identisk med den som er tilgjengelig ved hjelpe systemet i Splus, men gjengis her slik at leserne kan ha denne lett tilgjengelig. Dokumentasjonen for `glm` gir en nærmere beskrivelse av de ulike opsjonene denne kommandoen har. I dokumentasjonen for `glm.objekt` er det beskrevet hva som returneres ved bruk av `glm` kommandoen. Endelig angir `family` blant annet hvilke fordelingstyper `glm` tillater og hvilke linkfunksjoner som er tillatt for hver av disse.

glm

```
Fit a Generalized Linear Model


DESCRIPTION:
       Produces an object of class "glm" which is  a  generalized
       linear fit of the data.

USAGE:
       glm(formula, family = gaussian, data=<<see below>>,
           weights=<<see below>>, subset=<<see below>>, na.action=na.fail,
           start=<<see below>>, control, trace=F, model=F, x=F, y=T,
           contrasts=NULL, qr=F, ...)

REQUIRED ARGUMENTS:
formula:    a formula expression as for other regression  models,
       of  the  form response ~ predictors. See the documentation
       of lm and formula for details.

OPTIONAL ARGUMENTS:
family:   a family object - a list of functions  and  expressions
       for  defining the link and variance functions, initializa-
       tion and iterative weights. Families supported  are  gaus-
       sian,  binomial,  poisson,  Gamma,  inverse.gaussian  and
```

quasi. Functions like binomial produce a family object, but can be given without the parentheses. Family functions can take arguments, as in binomial(link=probit).

data: an optional data frame in which to interpret the variables occurring in the formula.

weights: the optional weights for the fitting criterion.

subset: expression saying which subset of the rows of the data should be used in the fit. This can be a logical vector (which is replicated to have length equal to the number of observations), or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.

na.action: a function to filter missing data. This is applied to the model.frame after any subset argument has been used. The default (with na.fail) is to create an error if any missing values are found. A possible alternative is na.omit, which deletes observations that contain one or more missing values.

start: a vector of initial values on the scale of the linear predictor.

control: a list of iteration and algorithmic constants. See glm.control for their names and default values. These can also be set as arguments to glm itself.

trace: logical flag: if TRUE, details of the iterations are printed. This can also be set in the control argument.

model: if TRUE, the model.frame is returned. If this argument is itself a model.frame, then the formula and data arguments are ignored, and model is used to define the model.

x: logical flag: if TRUE, the model.matrix is returned.

y: logical flag: if TRUE, the response variable is returned (default is TRUE).

contrasts: a list of contrasts to be used for some or all of the factors appearing as variables in the model formula. The names of the list should be the names of the corresponding variables, and the elements should either be contrast-type matrices (matrices with as many rows as levels of the factor and with columns linearly independent of each other and of a column of one's), or else they should be functions that compute such contrast matrices.

qr: logical flag: if TRUE, the QR decomposition of the model.matrix is returned.

...: control arguments may be given directly, see the control argument.

VALUE:

an object of class glm is returned, which inherits from lm. See glm.object for details.

DETAILS:

The output can be examined by print, summary, plot, and anova. Components can be extracted using predict, fitted, residuals, deviance, formula, and family. It can be modified using update. It has all the components of an lm object, with a few more. Other generic functions that have methods for glm objects are drop1, add1, step and preplot. Use glm.object for further details.

The response variable must conform with the definition of family, for example factor or binary data if family=binomial is declared.

The model is fit using Iterative Reweighted Least Squares (IRLS). The working response and iterative weights are computed using the functions contained in the family object. GLM models can also be fit using the function gam. The workhorse of glm is the function glm.fit which expects an x and y argument rather than a formula.

REFERENCES:

McCullagh, P. and Nelder, J. A. (1983). Generalized Linear Models. Chapman and Hall, London.

SEE ALSO:

glm.object, glm.control, gam, family, lm.

EXAMPLES:

```
glm(Count ~ ., data = solder, family = poisson)
glm(Kyphosis ~ poly(Age, 2) + (Number > 10)*Start,
    family = binomial)
glm(ozone^(1/3) ~ bs(radiation, 5) + poly(wind, temperature,
degree = 2), data=air)
```

glm.object

Generalized Linear Model Object

This class of objects is returned by the glm function to represent a fitted generalized linear model. Class glm inherits from class lm, since it is fit by iterative reweighted least squares; the object returned has all the components of a weighted least squares object. The class of gam objects, on the other hand, inherit from class glm.

Objects of this class have methods for the functions print, plot, summary, anova, predict, fitted, drop1, add1, and step, amongst others.

COMPONENTS:

The following components must be included in a legitimate glm object. The residuals, fitted values, coefficients and effects should be extracted by the generic functions of the same name, rather than by the "$" operator. The family function returns the entire family object used in the fitting, and deviance can be used to extract the deviance of the fit.

coefficients: the coefficients of the linear.predictors, which multiply the columns of the model matrix. The names of the coefficients are the names of the single-degree-of-freedom effects (the columns of the model matrix). If the model is over-determined there will be missing values in the coefficients corresponding to inestimable coefficients.

linear.predictors: the linear fit, given by the product of the model matrix and the coefficients; also the fitted.values from the final weighted least squares fit.

fitted.values: the fitted mean values, obtained by transforming linear.predictors using the inverse link function.

residuals: the residuals from the final weighted least squares fit; also known as working residuals, these are typically not interpretable without rescaling by the weights.

deviance: up to a constant, minus twice the maximized log-likelihood. Similar to the residual sum of squares.

null.deviance: the deviance corresponding to the model with no predictors.

iter: the number of IRLS iterations used to compute the estimates.

family: a 3 element character vector giving the name of the family, the link and the variance function; mainly for printing purposes.

weights: the iterative weights from the final IRLS fit

The object will also have the components of an lm object: coefficients, residuals, fitted.values, call, terms and some others involving the numerical fit. See lm.object.

SEE ALSO:

glm, gam.object, lm.object.

family

Generate a Family Object

USAGE:

```
family(object)
binomial(link=logit)
gaussian()
Gamma(link=inverse)
inverse.gaussian()
poisson(link=log)
quasi(link=identity, variance=constant)
```

ARGUMENTS:

link: the choices of link functions are logit, probit, cloglog, identity, inverse, log, "1/mu^2", and sqrt. Not all links are suitable for all families.

variance: the choices of variance functions are constant, mu(1-mu), mu, mu^2, and mu^3. This argument may be used only with quasi; each of the other families implies a variance function.

object: any object from which a family object can be extracted. Typically a fitted model object, with a default of gaussian.

VALUE:

a family object, which is a list of functions and expressions used by glm and gam in their iteratively reweighted least-squares algorithms. See family.object for details.

DETAILS:

Each of the names, except for quasi and the family extractor function family, are associated with a member of the exponential family of distributions. As such, they have a fixed variance function. There is typically a choice of link functions, with the default corresponding to the canonical link for that family. The quasi name represents Quasi-likelihood and need not correspond to any particular distribution; rather quasi can be used to combine any available link and variance function.

The following table summarizes the suitable pairings:

	binomial	gaussian	Gamma	inverse.gaussian	poisson	quasi
logit	*					*
probit	*					*
cloglog	*					*
identity		*	*		*	*
inverse			*			*
log			*		*	*
1/mu^2				*		*
sqrt					*	*

The function power can also be used to generate a power link function object for use with quasi; power takes an argument lambda.

Users can construct their own families, as long as they have compatible components having the same names as those, for example, of binomial. The easiest way is to use quasi with home-made link and variance objects; otherwise make.family can be used, or else direct construction of the family object. When passed as an argument to glm or gam with the default link, the empty parentheses () can be omitted. There is a print method for the class "family".

SEE ALSO:

family.object, glm, gam, robust, power.

EXAMPLES:

```
binomial(link = probit) # generate binomial family with probit link
glm(formula, family = binomial)
robust(gaussian) # create a robust version of the binomial family
gam(formula, family = robust(quasi(link = power(2))))  # the works!
```