Epidemiology with R

Bendix Carstensen Steno Diabetes Center
Gentofte, Denmark
http://BendixCarstensen.com

Université Bordeaux
23 January 2015

http://BendixCarstensen.com/Epi

Study types and data types

Epidemiology with R
23 January 2015
Université Bordeaux
http://BendixCarstensen.com/Epi

study-types

Epidemiological study types

- **Cross-sectional** studies:
  What is disease status at a particular date
- **Follow-up** studies:
  What is the rate of disease occurrence
  - Fixed cohorts, population based surveys
  - Dynamic cohorts
  - An entire population followed through registers
    Medical demography
- **Case-control** studies:
  Compare cases with non-cases.
  - Sampling based on disease status
  - Partial measures of disease occurrence/presence

Epidemiological data types

- **Continuous** (metric) responses can emerge from any observational design.
- **Categorical** response data essentially always **derived** from follow-up data:
  - Tables of counts from a cross-sectional study.
  - Tables of counts and follow-up time.
  - Tables of case-control status and exposure.
- Continuous and categorical explanatory variables occur in any design.

Cross-sectional studies

- What **fraction of the population** has a certain characteristic (such as a diagnosis of diabetes or other disease).
- **Observations:** the entire population (or a sample of it) classified by disease status
- The **likelihood** is a binomial likelihood for

$$p = \mathrm{P}\left\{\text{presence of disease}\right\}$$

- . . . that is, how p depends on explanatory variables.

Follow-up studies

- **Medical demography** — describing the entire population w.r.t. disease status over time
  - An entire population is followed for a particular event of interest (CVD, death, . . .)
- **Epidemiological** (observational) study
  Part of the population (a cohort) is followed for a limited period of time
  - May not necessarily be generalizable.
  - — but can elucidate the size of exposure effects on disease occurrence.
  - Neither exposures nor outcomes need be representative — only their relationship.

Follow-up studies

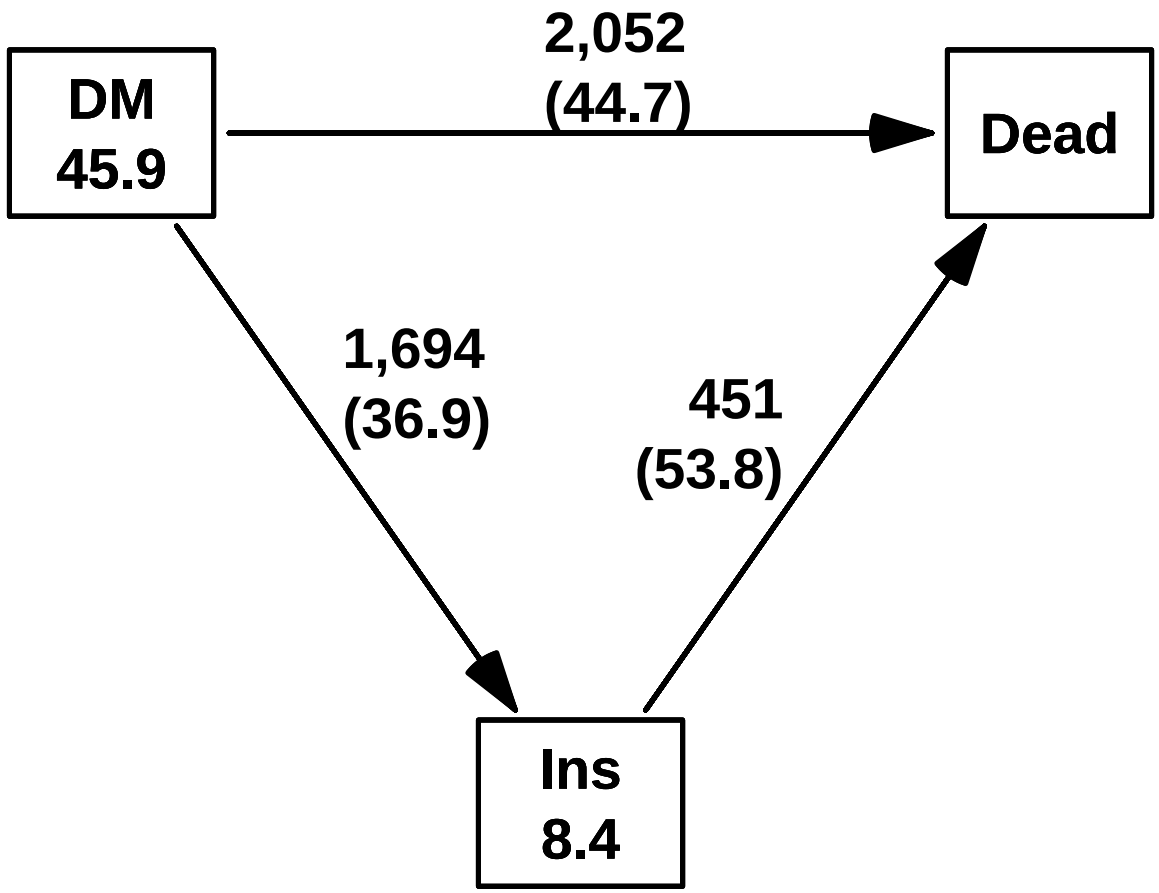
- **Observations** are (empirical) rates:
  (d, y): d events during y follow-up time
  (risk time, exposure time, person-years)
- **Models** for occurrence rates:

$$\lambda(t) = \mathrm{P}\left\{\text{event in } (t, t+h) \middle| \text{ no event till } t\right\}/h$$

- The **likelihood** for this is proportional to a Poisson likelihood (if λ is constant):

$$\text{log-lik} = \ell(\lambda|d, y) = d\log(\lambda) + \lambda y$$

How a follow-up study looks



Follow-up studies

- Each transition can be considered separately
- Rates modelled seperately (or jointly)
- Probabilities can be derived from estimated rates
- Simplest probability is:

$$S(t) = \mathrm{P}\left\{\text{survive till time } t\right\}$$

- Other probabilities of interest, *e.g.*:

$$P_c(t) = \mathrm{P}\left\{\text{die from cause } c \text{ before } t\right\}$$

— depends on more than one rate.

Case-control studies

- Events (cases) are sampled.
- But risk time is not. . .
  — it is replaced by a carefully chosen sample of the non-event persons.
- The likelihood is a binomial likelihood for

$$p = \mathrm{P}\left\{\text{case} \mid \text{included in the study}\right\}$$

  which contains the parameters of interest (and some not of any interest) *e.g.* rate-ratios.

Simple analyses of relationships

Epidemiology with R
23 January 2015
Université Bordeaux
http://BendixCarstensen.com/Epi

simp-ana

Variables in models

Response variables must be numeric.

- Metric (a measurement with units)
- Binary (two values coded 0/1)
- Failure (does the subject fail at end of follow-up) and FU-time
- Count (aggregated failure data)

Explanatory variables can be:

- Numeric
- Factor (classes)

Simple analyses

- Response as a function of exposure:

$$y = \mu + \beta x$$

- — **controlled** for levels of a confounder:

$$y = \mu + \beta x + \delta_c$$

confounding refers to the **change** in β

- — **stratified** by levels of an effect modifier:

$$y = \mu + \beta_e x + \gamma_e$$

effect modification is called **interaction** in statistics

Simple analyses made simple: `effx`

```
effx( response, type = "metric",
                   fup = NULL,
              exposure,
                strata = NULL,
               control = NULL,
               weights = NULL,
                 alpha = 0.05,
                  base = 1,
                digits = 3,
                  data = NULL )
```

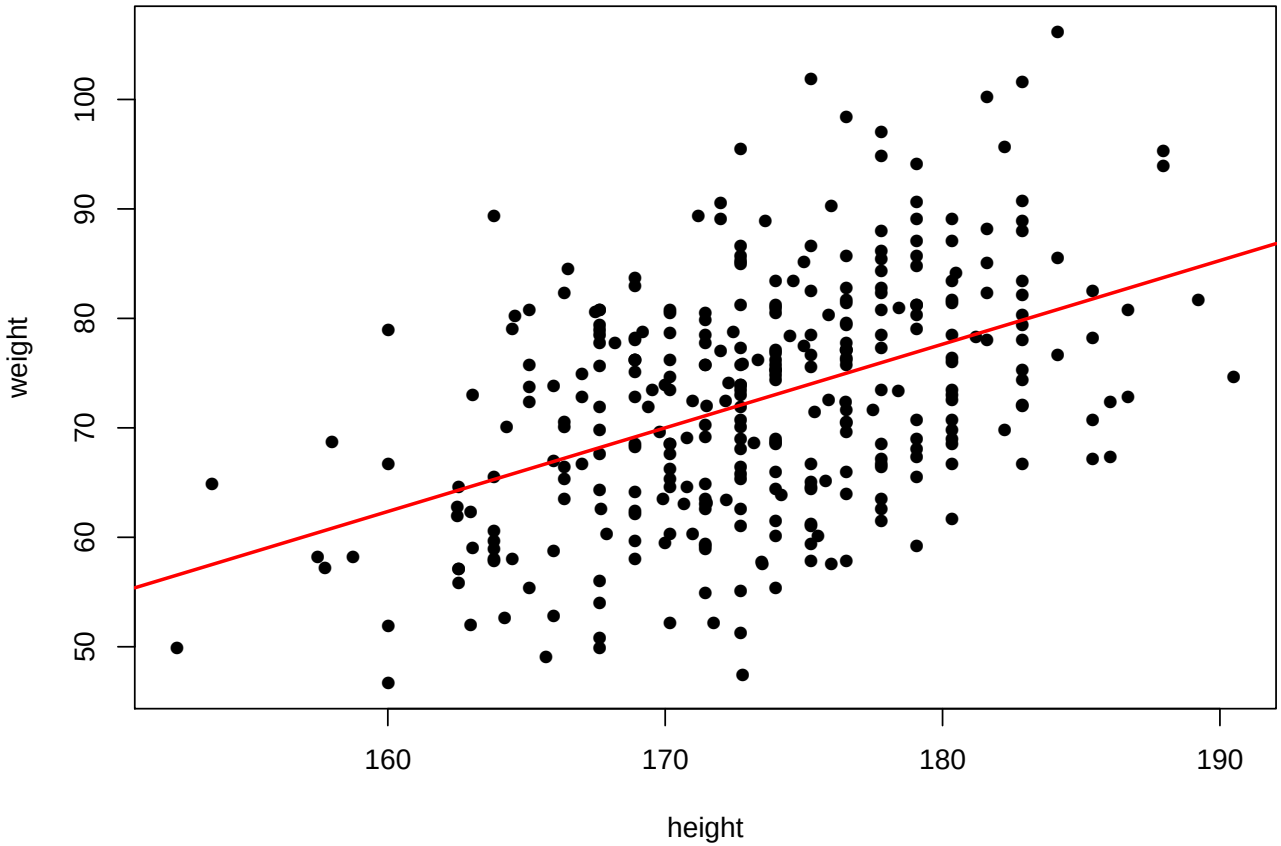
— so let's see how it works; exercise

Linear relationships

Epidemiology with R
23 January 2015
Université Bordeaux
http://BendixCarstensen.com/Epi

Linear models

```
> options( show.signif.stars=FALSE, width=60 )
> library( Epi )
> data( diet )
> names( diet )

 [1] "id"         "doe"        "dox"        "dob"
 [5] "y"          "fail"       "job"        "month"
 [9] "energy"     "height"     "weight"     "fat"
[13] "fibre"      "energy.grp" "chd"

> with( diet, plot(weight~height,pch=16) )
> abline( lm(weight~height,data=diet), col="red", lwd=2 )
```




```
> with( diet, plot(weight~height,pch=16) )
> abline( lm(weight~height,data=diet), col="red", lwd=2 )
```

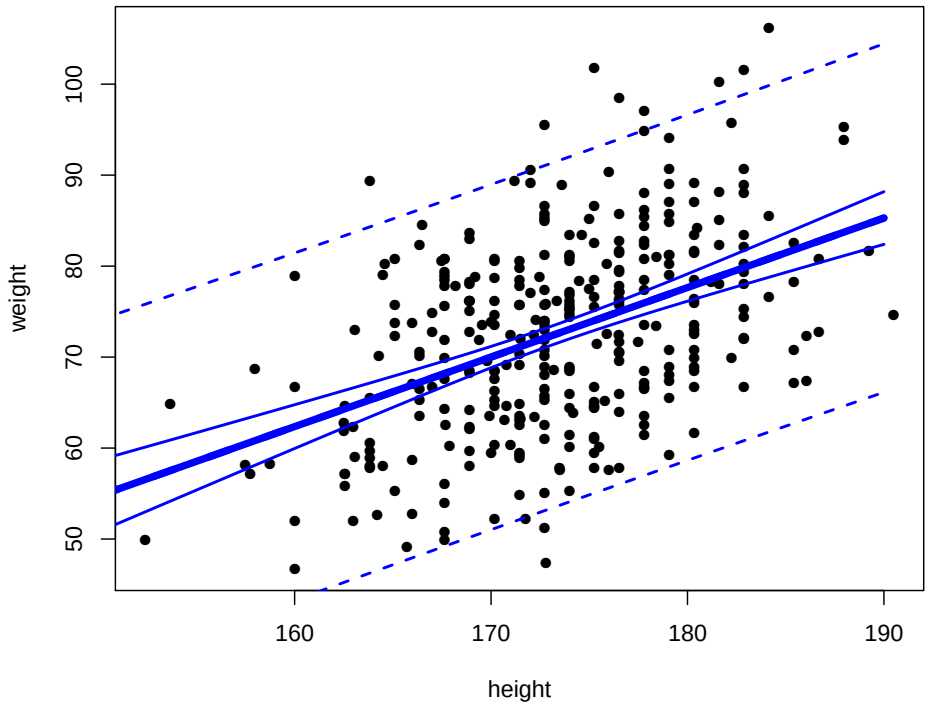
Linear models, extracting estimates

```
> ml <- lm( weight ~ height, data=diet )
```

```
> summary( ml )
```

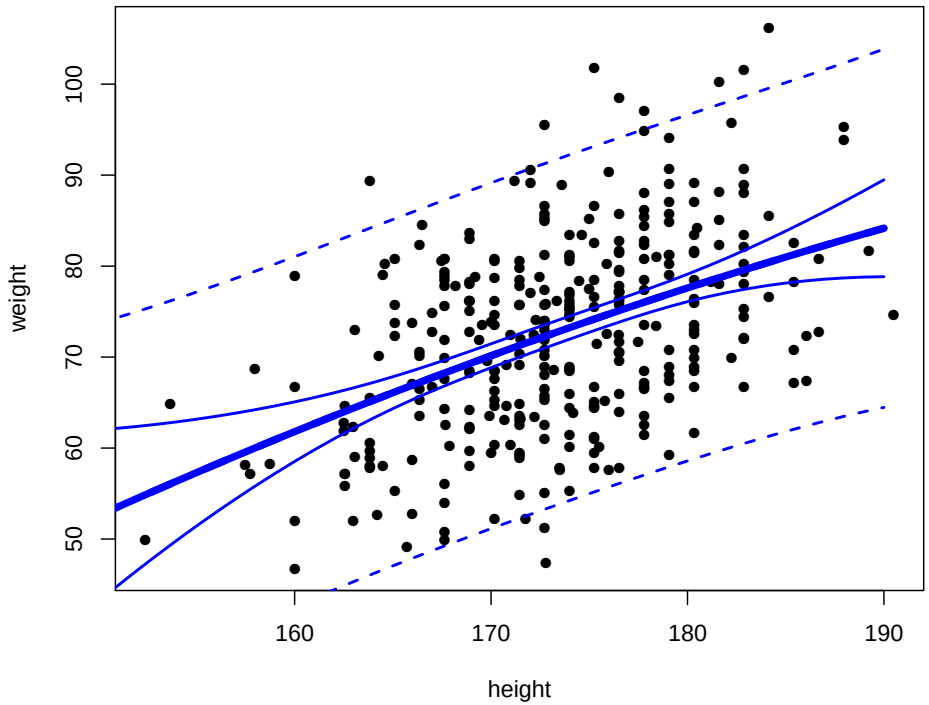
```
Call:
lm(formula = weight ~ height, data = diet)

Residuals:
     Min       1Q   Median       3Q      Max
-24.7361  -7.4553   0.1608   6.9384  27.8130

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -59.91601   14.31557  -4.185 3.66e-05
height        0.76421    0.08252   9.261  < 2e-16

Residual standard error: 9.625 on 330 degrees of freedom
  (5 observations deleted due to missingness)
Multiple R-squared:  0.2063, Adjusted R-squared:  0.2039
F-statistic: 85.76 on 1 and 330 DF,  p-value: < 2.2e-16
```

Linear models, prediction




```
> ml <- lm( weight ~ height, data=diet )
> nd <- data.frame( height = 150:190 )
> pr.co <- predict( ml, newdata=nd, interval="conf" )
> pr.pr <- predict( ml, newdata=nd, interval="pred" )
> with( diet, plot( weight ~ height, pch=16 )  )
> matlines( nd$height, pr.co, lty=1, lwd=c(5,2,2), col="blue" )
> matlines( nd$height, pr.pr, lty=2, lwd=c(5,2,2), col="blue" )
```

non-Linear models, prediction




```
> mq <- lm( weight ~ height + I(height^2), data=diet )
> nd <- data.frame( height = 150:190 )
> pr.co <- predict( mq, newdata=nd, interval="conf" )
> pr.pr <- predict( mq, newdata=nd, interval="pred" )
> with( diet, plot( weight ~ height, pch=16 )  )
> matlines( nd$height, pr.co, lty=1, lwd=c(5,2,2), col="blue" )
> matlines( nd$height, pr.pr, lty=2, lwd=c(5,2,2), col="blue" )
```

Curved relationships

Epidemiology with R
23 January 2015
Université Bordeaux
http://BendixCarstensen.com/Epi

Testis cancer

Testis cancer in Denmark:

```
> options( show.signif.stars=FALSE )
> library( Epi )
> data( testisDK )
> str( testisDK )

  'data.frame': 4860 obs. of  4 variables:
   $ A: num  0 1 2 3 4 5 6 7 8 9 ...
   $ P: num  1943 1943 1943 1943 1943 ...
   $ D: num  1 1 0 1 0 0 0 0 0 0 ...
   $ Y: num  39650 36943 34588 33267 32614 ...

> head( testisDK )

    A    P D        Y
  1 0 1943 1 39649.50
  2 1 1943 1 36942.83
  3 2 1943 0 34588.33
  4 3 1943 1 33267.00
  5 4 1943 0 32614.00
  6 5 1943 0 32020.33
```

Cases, PY and rates

```
> stat.table( list(A=floor(A/10)*10,
+                  P=floor(P/10)*10),
+             list( D=sum(D),
+                   Y=sum(Y/1000),
+                rate=ratio(D,Y,10^5) ),
+             margins=TRUE, data=testisDK )
```

	P					
A	1940	1950	1960	1970	1980	1990
0	10.00	7.00	16.00	18.00	9.00	10.00
	2604.66	4037.31	3884.97	3820.88	3070.87	2165.54
	0.38	0.17	0.41	0.47	0.29	0.46
10	13.00	27.00	37.00	72.00	97.00	75.00
	2135.73	3505.19	4004.13	3906.08	3847.40	2260.97
	0.61	0.77	0.92	1.84	2.52	3.32
20	124.00	221.00	280.00	535.00	724.00	557.00
	2225.55	2923.22	3401.65	4028.57	3941.18	2824.58
	5.57	7.56	8.23	13.28	18.37	19.72

Linear effects in `glm`

How do rates depend on age?

```
> ml <- glm( D ~ A, offset=log(Y), family=poisson, data=testisDK
> round( ci.lin( ml ), 4 )

             Estimate StdErr          z P    2.5%   97.5%
(Intercept)  -9.7755 0.0207 -472.3164 0 -9.8160 -9.7349
A             0.0055 0.0005   11.3926 0  0.0045  0.0064

> round( ci.exp( ml ), 4 )

             exp(Est.)   2.5%  97.5%
(Intercept)     0.0001 0.0001 0.0001
A               1.0055 1.0046 1.0064
```

Linear increase of log-rates by age

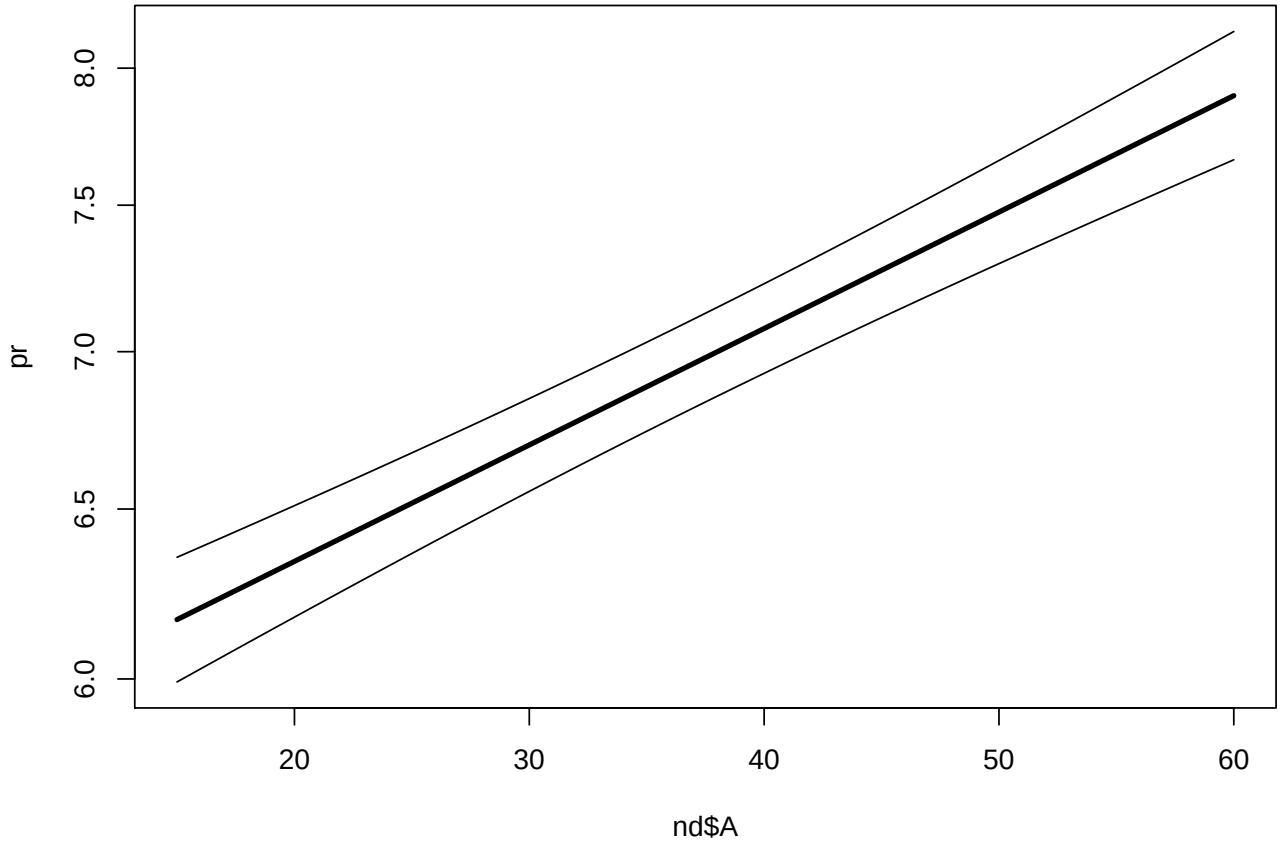
Linear effects in glm

```
> nd <- data.frame( A=15:60, Y=10^5 )
> pr <- ci.pred( m1, newdata=nd )
> head( pr )

   Estimate     2.5%    97.5%
1 6.170105 5.991630 6.353896
2 6.204034 6.028525 6.384652
3 6.238149 6.065547 6.415662
4 6.272452 6.102689 6.446937
5 6.306943 6.139944 6.478485
6 6.341624 6.177301 6.510319

> matplot( nd$A, pr,
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Linear effects in glm

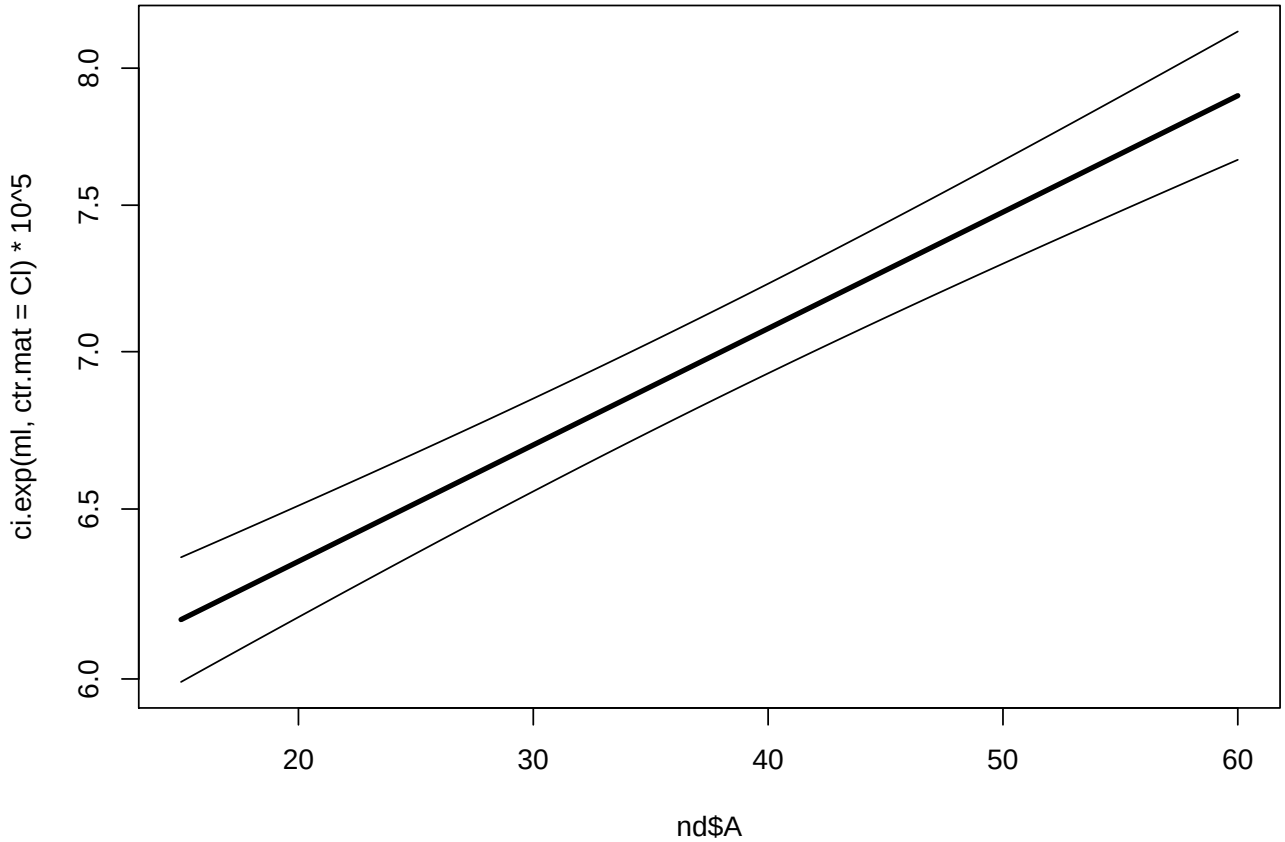
```
> round( ci.lin( m1 ), 4 )

             Estimate StdErr          z P    2.5%   97.5%
(Intercept)  -9.7755 0.0207 -472.3164 0 -9.8160 -9.7349
A             0.0055 0.0005   11.3926 0  0.0045  0.0064

> Cl <- cbind( 1, nd$A )
> head( Cl )

     [,1] [,2]
[1,]    1   15
[2,]    1   16
[3,]    1   17
[4,]    1   18
[5,]    1   19
[6,]    1   20

> matplot( nd$A, ci.exp( m1, ctr.mat=Cl ),
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Linear effects in `glm`




```
> matplot( nd$A, pr,
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Linear effects in `glm`




```
> matplot( nd$A, ci.exp( ml, ctr.mat=Cl )*10^5,
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Quadratic effects in `glm`

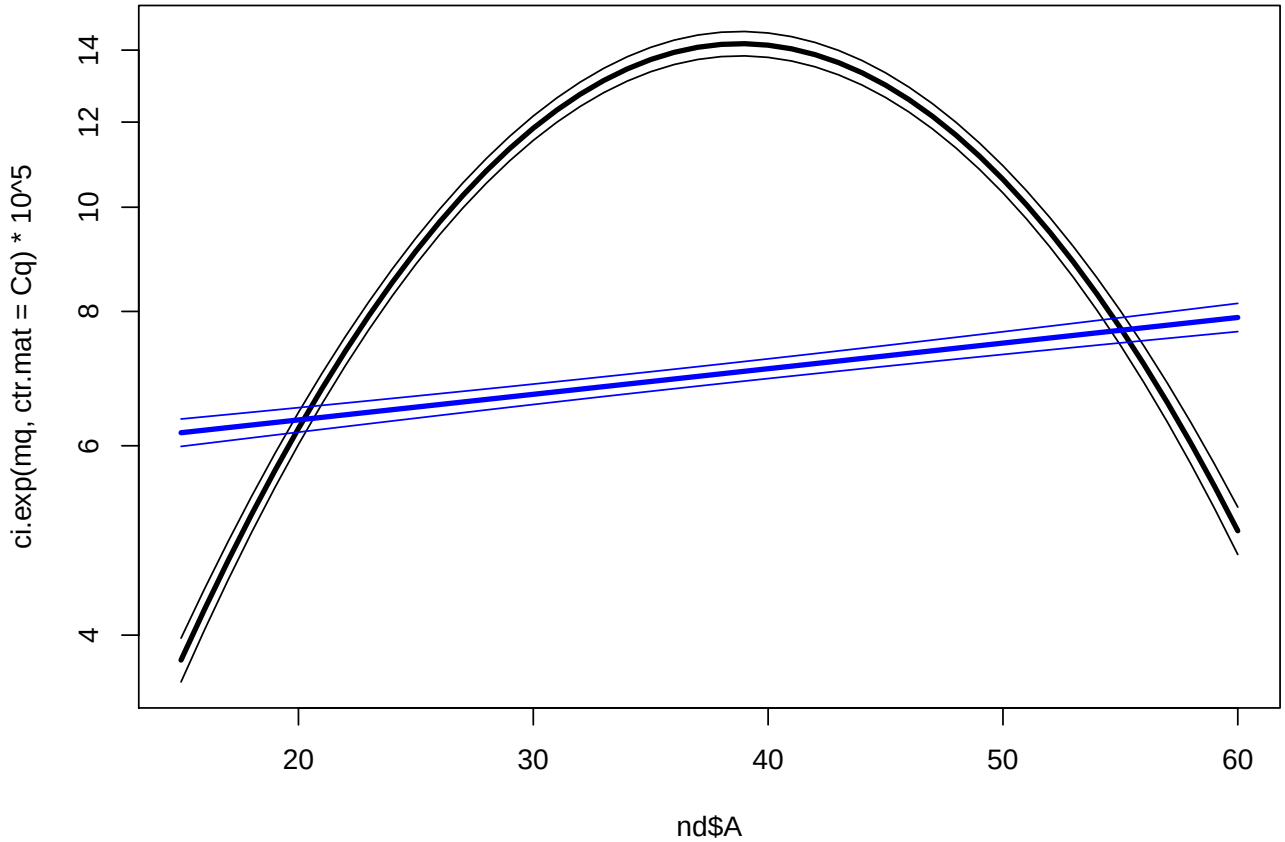
How do rates depend on age?

```
> mq <- glm( D ~ A + I(A^2),
+            offset=log(Y), family=poisson, data=testisDK )
> round( ci.lin( mq ), 4 )

              Estimate StdErr         z P     2.5%    97.5%
(Intercept) -12.3656 0.0596 -207.3611 0 -12.4825 -12.2487
A             0.1806 0.0033   54.8290 0   0.1741   0.1871
I(A^2)       -0.0023 0.0000  -53.7006 0  -0.0024  -0.0022

> round( ci.exp( mq ), 4 )

              exp(Est.)   2.5%  97.5%
(Intercept)      0.0000 0.0000 0.0000
A                1.1979 1.1902 1.2057
I(A^2)           0.9977 0.9976 0.9978
```

Quadratic effect in glm

```
> round( ci.lin( mq ), 4 )

              Estimate StdErr         z P     2.5%    97.5%
(Intercept) -12.3656 0.0596 -207.3611 0 -12.4825 -12.2487
A             0.1806 0.0033   54.8290 0   0.1741   0.1871
I(A^2)       -0.0023 0.0000  -53.7006 0  -0.0024  -0.0022

> Cq <- cbind( 1, 15:60, (15:60)^2 )
> head( Cq, 4 )

     [,1] [,2] [,3]
[1,]    1   15  225
[2,]    1   16  256
[3,]    1   17  289
[4,]    1   18  324

> matplot( nd$A, ci.exp( mq, ctr.mat=Cq )*10^5,
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
```

Quadratic effect in `glm`




```
> matplot( nd$A, ci.exp( mq, ctr.mat=Cq )*10^5,
+          type="l", lty=1, lwd=c(3,1,1), col="black", log="y" )
> matlines( nd$A, ci.exp( ml, ctr.mat=Cl )*10^5,
+           type="l", lty=1, lwd=c(3,1,1), col="blue" )
```

Spline effects in glm

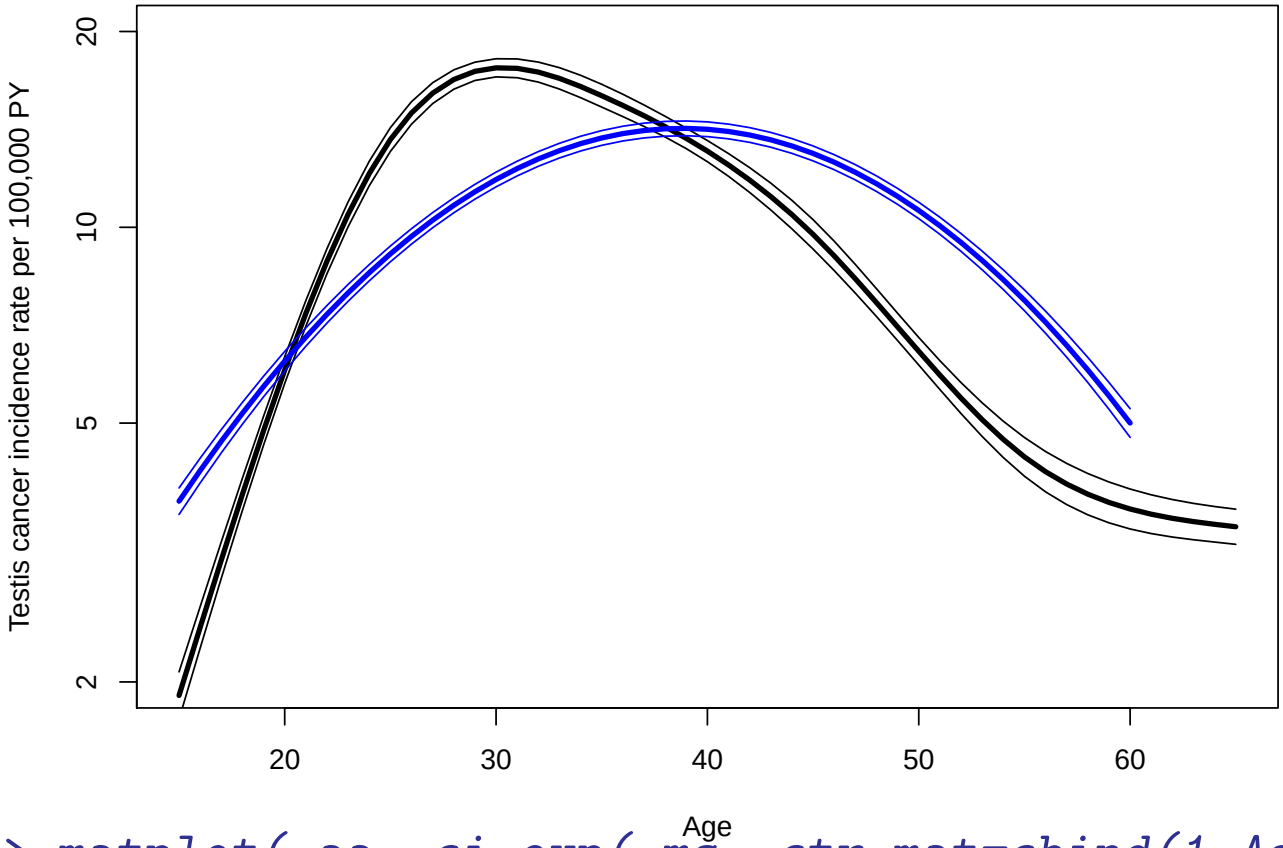
```
> library( splines )
> ms <- glm( D ~ Ns(A,knots=seq(15,65,10)),
+                  offset=log(Y), family=poisson, data=testisDK )
> round( ci.exp( ms ), 3 )
```

```
                                        exp(Est.)   2.5%  97.5%
(Intercept)                                 0.000  0.000  0.000
Ns(A, knots = seq(15, 65, 10))1             8.548  7.650  9.551
Ns(A, knots = seq(15, 65, 10))2             5.706  4.998  6.514
Ns(A, knots = seq(15, 65, 10))3             1.002  0.890  1.128
Ns(A, knots = seq(15, 65, 10))4            14.402 11.896 17.436
Ns(A, knots = seq(15, 65, 10))5             0.466  0.429  0.505
```

```
> aa <- 15:65
> As <- Ns( aa, knots=seq(15,65,10) )
> head( As )
```

```
                  1 2           3           4           5
[1,] 0.0000000000 0  0.00000000 0.00000000  0.00000000
[2,] 0.0001666667 0 -0.02527011 0.07581034 -0.05054022
[3,] 0.0013333333 0 -0.05003313 0.15009940 -0.10006626
[4,] 0.0045000000 0 -0.07378197 0.22134590 -0.14756393
[5,] 0.0106666667 0 -0.09600952 0.28802857 -0.19201905
[6,] 0.0208333333 0 -0.11620871 0.34862613 -0.23241742
```

Spline effects in `glm`




```
> matplot( aa, ci.exp( ms, ctr.mat=cbind(1,As) )*10^5,
+          log="y", xlab="Age", ylab="Testis cancer incidence ra
+          type="l", lty=1, lwd=c(3,1,1), col="black", ylim=c(2,
> matlines( nd$A, ci.exp( mq, ctr.mat=Cq )*10^5,
+           type="l", lty=1, lwd=c(3,1,1), col="blue" )
```

Adding a linear period effect

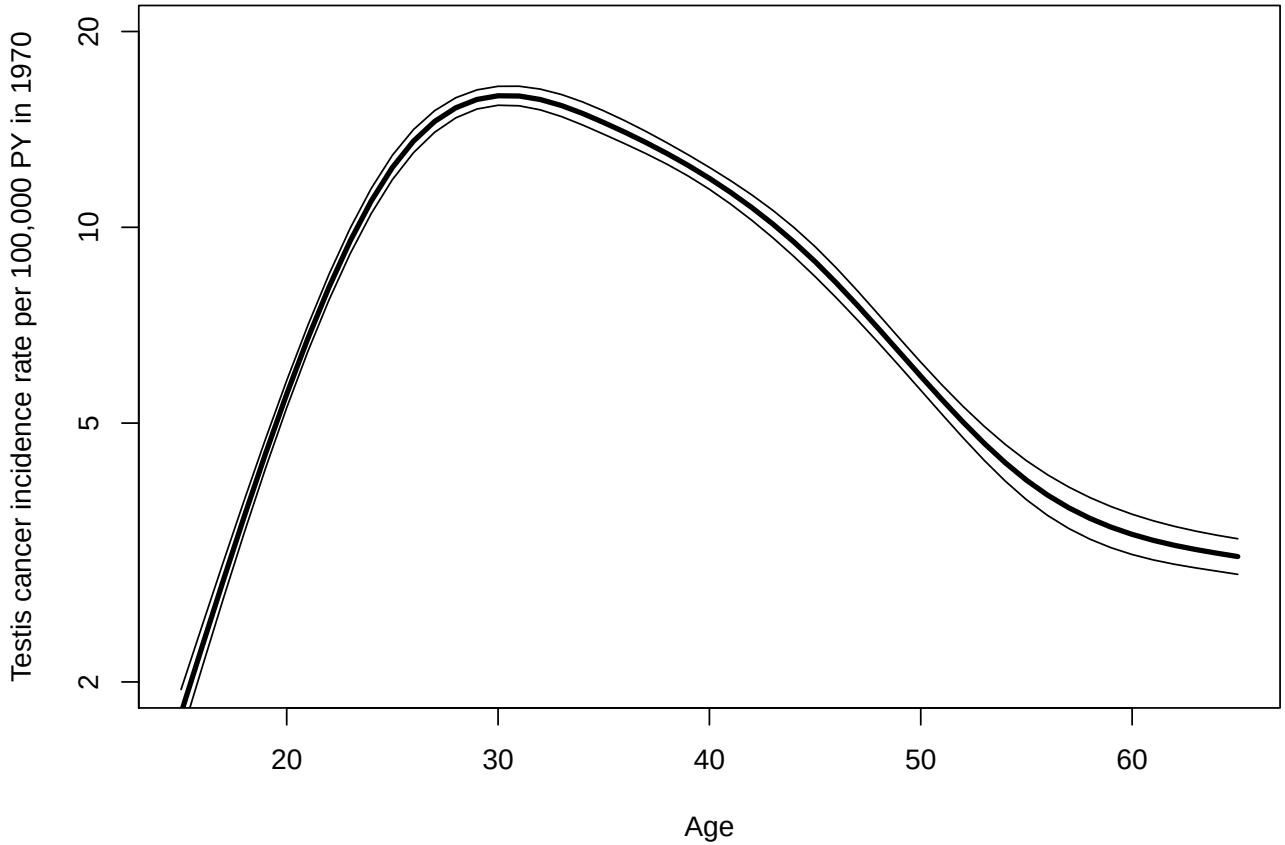
```
> msp <- glm( D ~ Ns(A,knots=seq(15,65,10)) + P,
+                 offset=log(Y), family=poisson, data=testisDK )
> round( ci.lin( msp ), 3 )
```

	Estimate	StdErr	z	P
(Intercept)	-58.105	1.444	-40.229	0.000
Ns(A, knots = seq(15, 65, 10))1	2.120	0.057	37.444	0.000
Ns(A, knots = seq(15, 65, 10))2	1.700	0.068	25.157	0.000
Ns(A, knots = seq(15, 65, 10))3	0.007	0.060	0.110	0.913
Ns(A, knots = seq(15, 65, 10))4	2.596	0.097	26.631	0.000
Ns(A, knots = seq(15, 65, 10))5	-0.780	0.042	-18.748	0.000
P	0.024	0.001	32.761	0.000

```
> Ca <- cbind( 1, Ns( aa, knots=seq(15,65,10) ), 1970 )
> head( Ca )
```

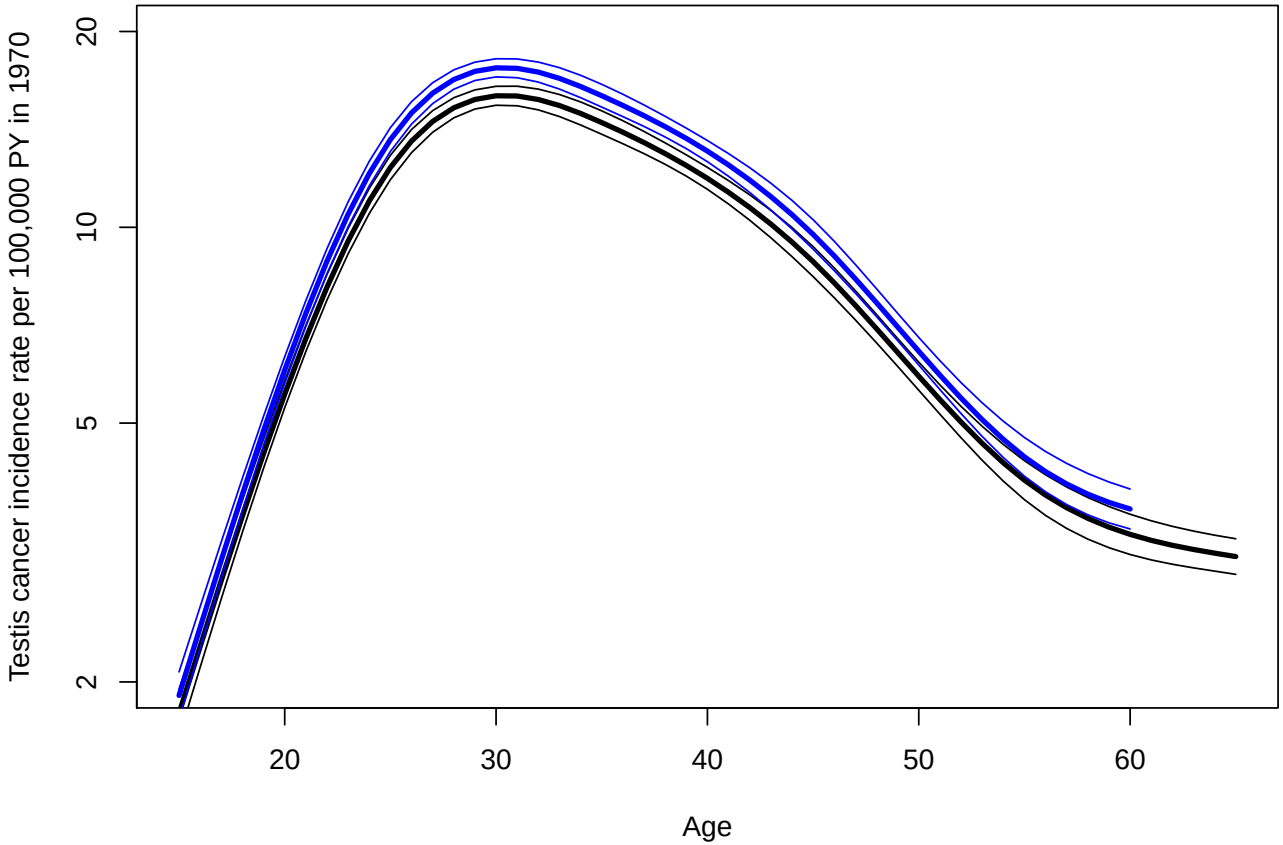
[1,]	1	0.0000000000	0	0.00000000	0.00000000	0.00000000	1970
[2,]	1	0.0001666667	0	-0.02527011	0.07581034	-0.05054022	1970
[3,]	1	0.0013333333	0	-0.05003313	0.15009940	-0.10006626	1970
[4,]	1	0.0045000000	0	-0.07378197	0.22134590	-0.14756393	1970
[5,]	1	0.0106666667	0	-0.09600952	0.28802857	-0.19201905	1970
[6,]	1	0.0208333333	0	-0.11620871	0.34862613	-0.23241742	1970

Adding a linear period effect




```
> matplot( aa, ci.exp( msp, ctr.mat=Ca )*10^5,
+          log="y", xlab="Age",
+          ylab="Testis cancer incidence rate per 100,000 PY in
+          type="l", lty=1, lwd=c(3,1,1), col="black", ylim=c(2,
```

Adding a linear period effect




```
> matplot( aa, ci.exp( msp, ctr.mat=Ca )*10^5,
+          log="y", xlab="Age",
+          ylab="Testis cancer incidence rate per 100,000 PY in
+          type="l", lty=1, lwd=c(3,1,1), col="black", ylim=c(2,
> matlines( nd$A, ci.pred( ms, newdata=nd ),
+           type="l", lty=1, lwd=c(3,1,1), col="blue" )
```

The period effect

```
> round( ci.lin( msp ), 3 )
```

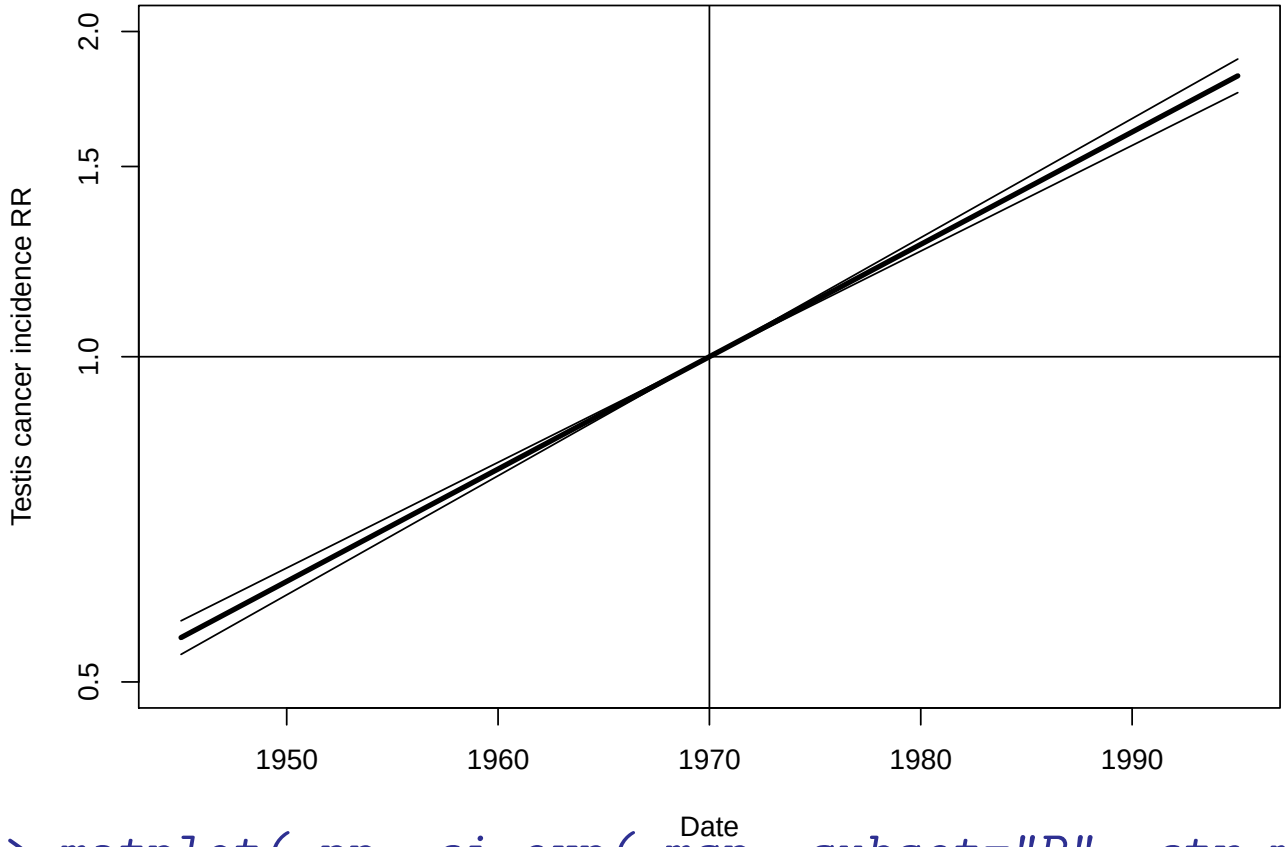
	Estimate	StdErr	z	P
(Intercept)	-58.105	1.444	-40.229	0.000
Ns(A, knots = seq(15, 65, 10))1	2.120	0.057	37.444	0.000
Ns(A, knots = seq(15, 65, 10))2	1.700	0.068	25.157	0.000
Ns(A, knots = seq(15, 65, 10))3	0.007	0.060	0.110	0.913
Ns(A, knots = seq(15, 65, 10))4	2.596	0.097	26.631	0.000
Ns(A, knots = seq(15, 65, 10))5	-0.780	0.042	-18.748	0.000
P	0.024	0.001	32.761	0.000

```
> pp <- seq(1945,1995,0.2)
> Cp <- cbind( pp ) - 1970
> head( Cp )
```

```
         pp
[1,] -25.0
[2,] -24.8
[3,] -24.6
[4,] -24.4
[5,] -24.2
[6,] -24.0
```

```
> ci.exp( msp, subset="P" )
```

Period effect




```
> matplot( pp, ci.exp( msp, subset="P", ctr.mat=Cp ),
+          log="y", ylim=c(0.5,2), xlab="Date",
+          ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1, v=1970 )
```

A quadratic period effect

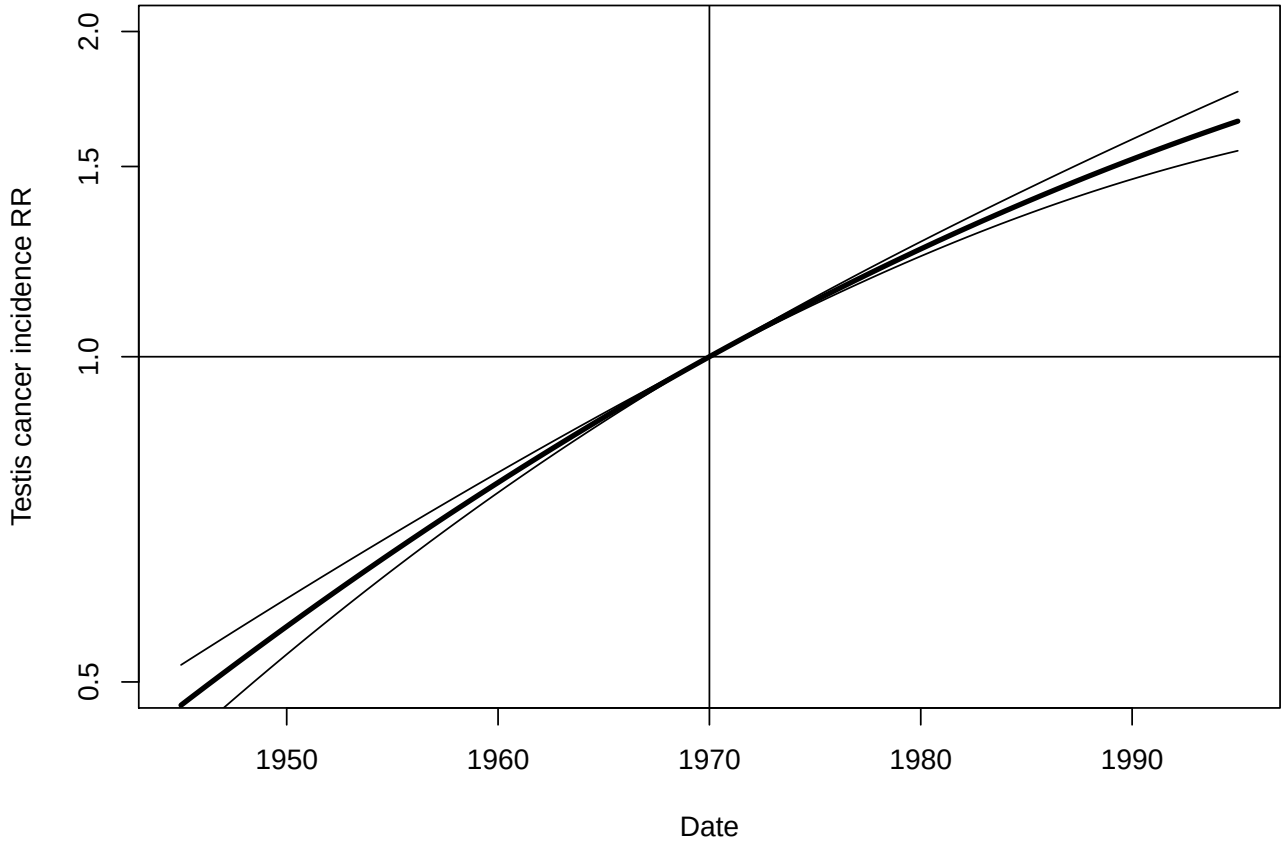
```
> mspq <- glm( D ~ Ns(A,knots=seq(15,65,10)) + P + I(P^2),
+                  offset=log(Y), family=poisson, data=testisDK
> round( ci.exp( mspq ), 3 )
```

	exp(Est.)	2.5%	97.5%
(Intercept)	0.000	0.000	0.000
Ns(A, knots = seq(15, 65, 10))1	8.356	7.478	9.337
Ns(A, knots = seq(15, 65, 10))2	5.513	4.829	6.295
Ns(A, knots = seq(15, 65, 10))3	1.006	0.894	1.133
Ns(A, knots = seq(15, 65, 10))4	13.439	11.101	16.269
Ns(A, knots = seq(15, 65, 10))5	0.458	0.422	0.497
P	2.189	1.457	3.291
I(P^2)	1.000	1.000	1.000

```
> Cq <- cbind( pp-1970, pp^2-1970^2 )
> head( Cq )
```

```
      [,1]      [,2]
[1,] -25.0 -97875.00
[2,] -24.8 -97096.96
[3,] -24.6 -96318.84
[4,] -24.4 -95540.64
[5,] -24.2 -94762.36
[6,] -24.0 -93984.00
```

A quadratic period effect




```
> matplot( pp, ci.exp( mspq, subset="P", ctr.mat=Cq ),
+          log="y", ylim=c(0.5,2), xlab="Date",
+          ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1, v=1970 )
```

A spline period effect

```
> msps <- glm( D ~ Ns(A,knots=seq(15,65,10)) +
+                  Ns(P,knots=seq(1950,1990,10)),
+                  offset=log(Y), family=poisson, data=testisDK
> round( ci.exp( msps ), 3 )
```

	exp(Est.)	2.5%	97.5%
(Intercept)	0.000	0.000	0.000
Ns(A, knots = seq(15, 65, 10))1	8.327	7.452	9.305
Ns(A, knots = seq(15, 65, 10))2	5.528	4.842	6.312
Ns(A, knots = seq(15, 65, 10))3	1.007	0.894	1.133
Ns(A, knots = seq(15, 65, 10))4	13.447	11.107	16.279
Ns(A, knots = seq(15, 65, 10))5	0.458	0.422	0.497
Ns(P, knots = seq(1950, 1990, 10))1	1.711	1.526	1.918
Ns(P, knots = seq(1950, 1990, 10))2	2.190	2.028	2.364
Ns(P, knots = seq(1950, 1990, 10))3	3.222	2.835	3.661
Ns(P, knots = seq(1950, 1990, 10))4	2.299	2.149	2.459

A spline period effect

```
> Cs <- Ns(                    pp  ,knots=seq(1950,1990,10))
> Cr <- Ns(rep(1970,length(pp)),knots=seq(1950,1990,10))
> head( Cs, 4 )
```

```
     1         2          3         4
[1,] 0 0.1267731 -0.3803194 0.2535463
[2,] 0 0.1217022 -0.3651066 0.2434044
[3,] 0 0.1166313 -0.3498939 0.2332626
[4,] 0 0.1115604 -0.3346811 0.2231207
```

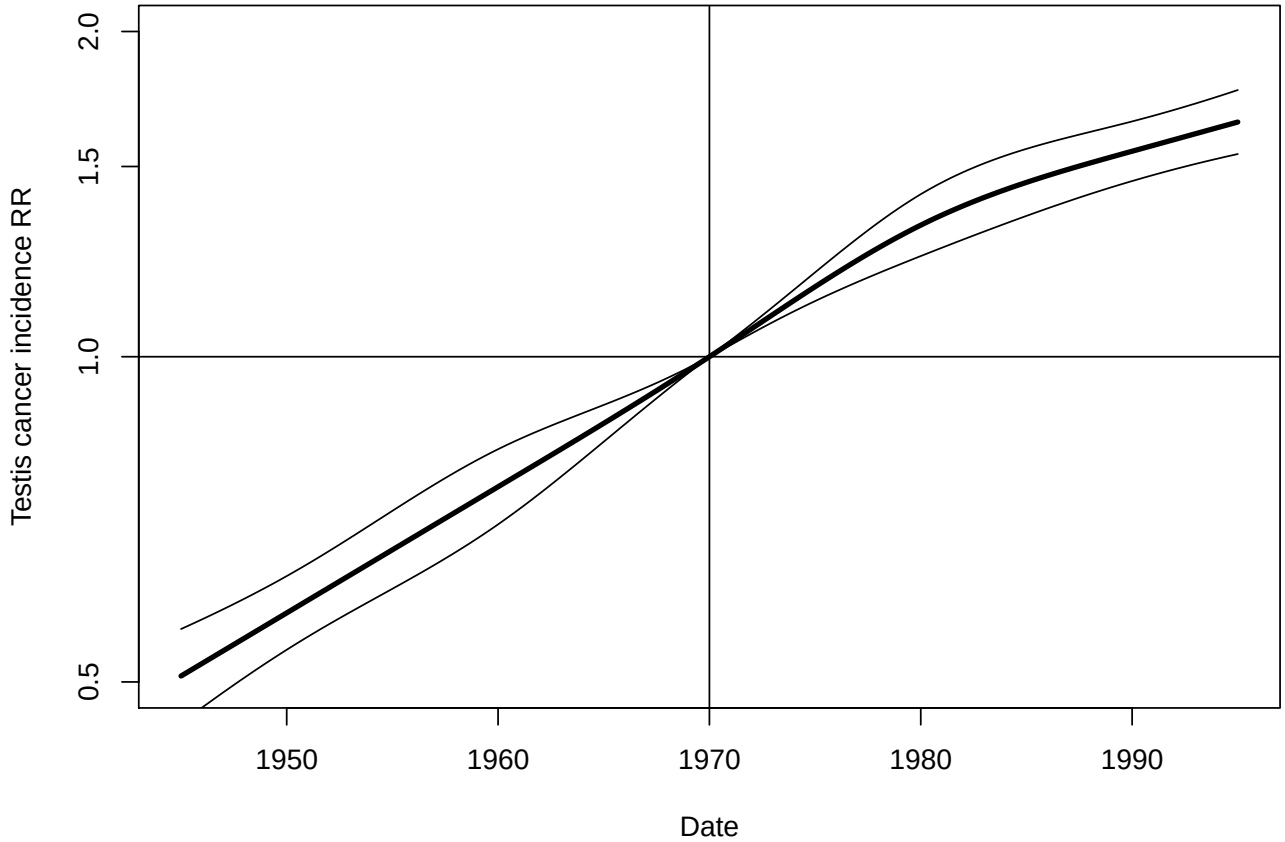
```
> head( Cr, 4 )
```

```
             1         2         3          4
[1,] 0.6666667 0.1125042 0.1624874 -0.1083249
[2,] 0.6666667 0.1125042 0.1624874 -0.1083249
[3,] 0.6666667 0.1125042 0.1624874 -0.1083249
[4,] 0.6666667 0.1125042 0.1624874 -0.1083249
```

```
> ci.exp( msps, subset="P" )
```

```
                                          exp(Est.)     2.5%    97.5
Ns(P, knots = seq(1950, 1990, 10))1  1.710808 1.525946 1.91806
Ns(P, knots = seq(1950, 1990, 10))2  2.189650 2.027898 2.36430
Ns(P, knots = seq(1950, 1990, 10))3  3.221563 2.835171 3.66061
```

Period effect

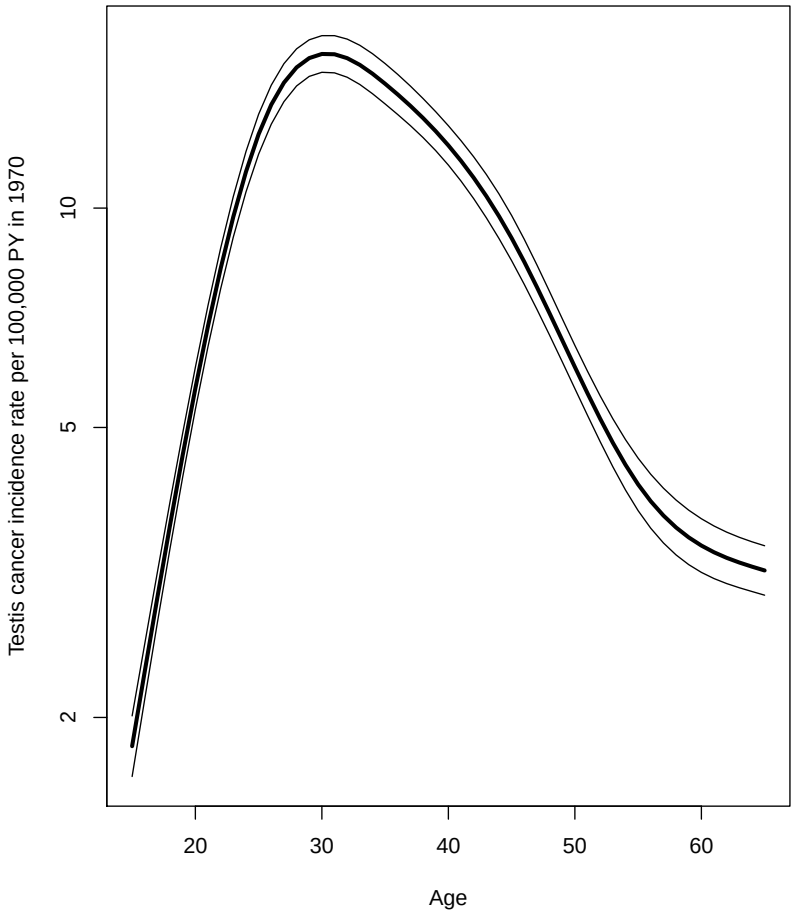


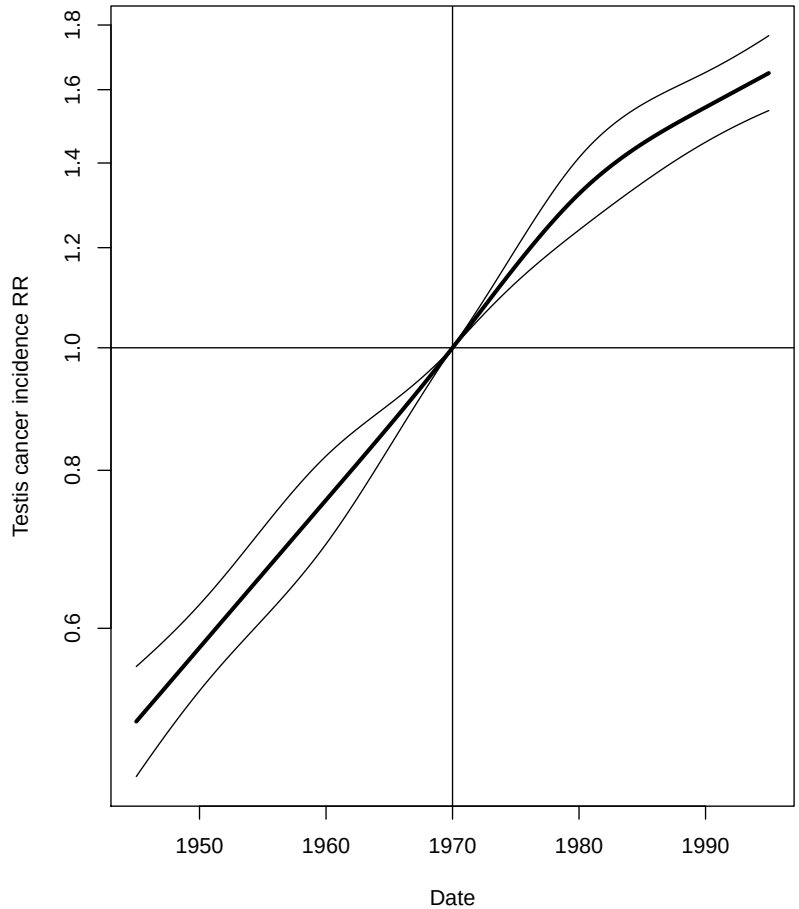

```
> matplot( pp, ci.exp( msps, subset="P", ctr.mat=Cs-Cr ),
+          log="y", ylim=c(0.5,2), xlab="Date",
+          ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1, v=1970 )
```

Period effect

```
> par( mfrow=c(1,2) )
> Cap <- cbind( 1, Ns(                    aa  ,knots=seq(15,65,10)),
+                  Ns(rep(1970,length(aa)),knots=seq(1950,1990,1
> matplot( aa, ci.exp( msps, ctr.mat=Cap )*10^5,
+          log="y", xlab="Age",
+          ylab="Testis cancer incidence rate per 100,000 PY in
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> matplot( pp, ci.exp( msps, subset="P", ctr.mat=Cs-Cr ),
+          log="y", xlab="Date", ylab="Testis cancer incidence R
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1, v=1970 )
```

Age and period effect

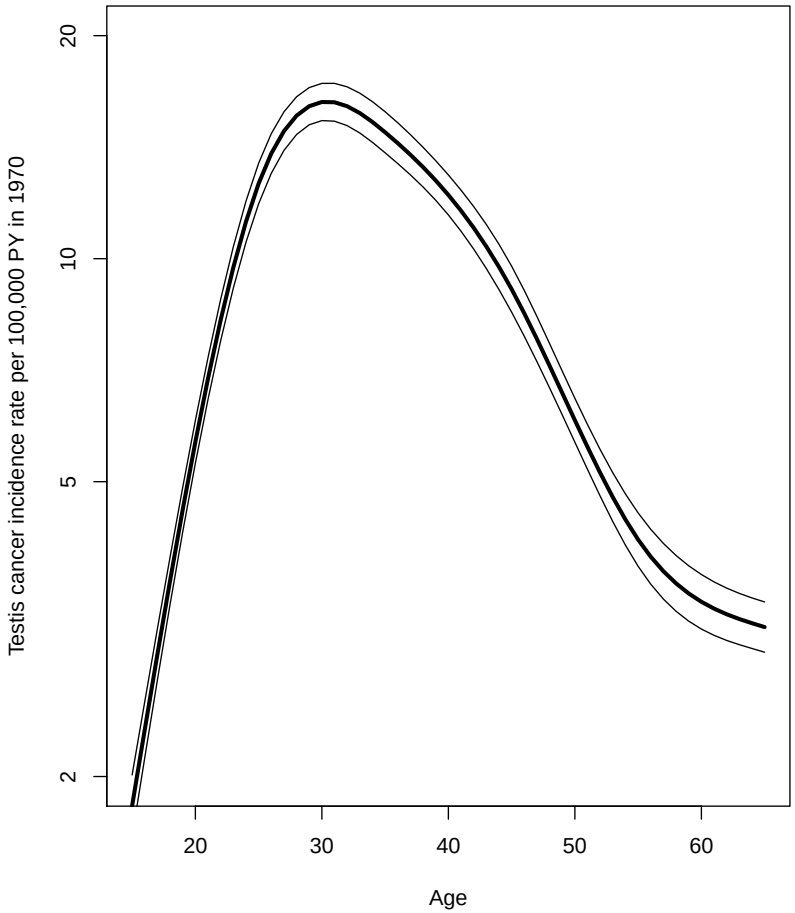


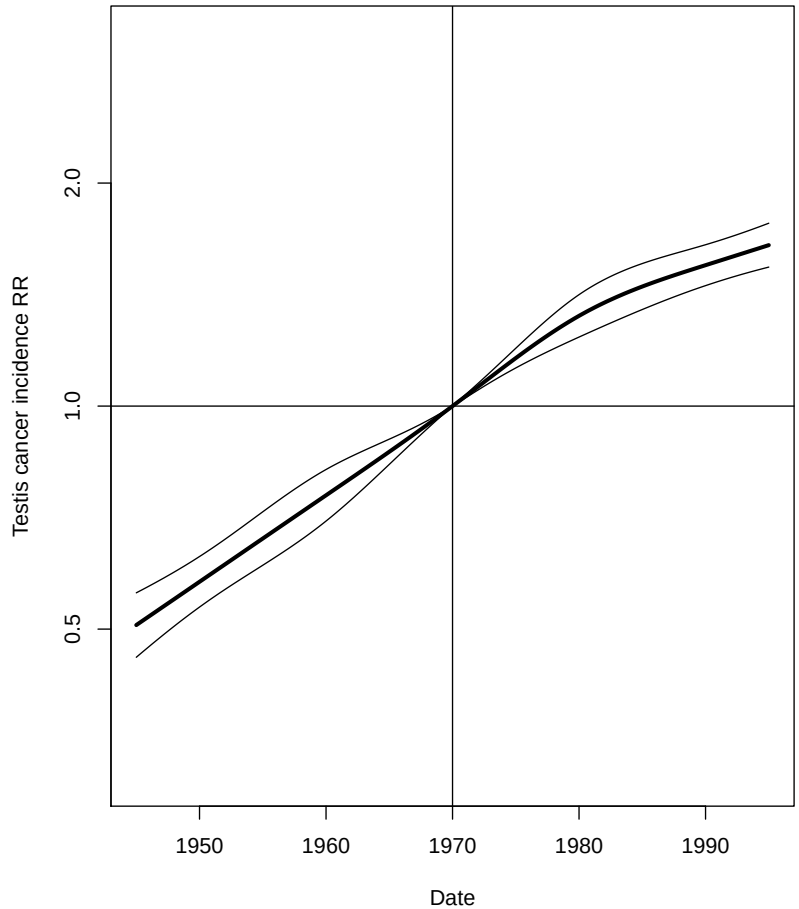


Period effect

```
> par( mfrow=c(1,2) )
> Cap <- cbind( 1, Ns(                    aa  ,knots=seq(15,65,10)),
+                  Ns(rep(1970,length(aa)),knots=seq(1950,1990,1
> matplot( aa, ci.exp( msps, ctr.mat=Cap )*10^5,
+          log="y", xlab="Age",
+          ylim=c(2,20), xlim=c(15,65),
+          ylab="Testis cancer incidence rate per 100,000 PY in
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> matplot( pp, ci.exp( msps, subset="P", ctr.mat=Cs-Cr ),
+          log="y", xlab="Date",
+          ylim=c(2,20)/sqrt(2*20), xlim=c(15,65)+1930,
+          ylab="Testis cancer incidence RR",
+          type="l", lty=1, lwd=c(3,1,1), col="black" )
> abline( h=1, v=1970 )
```

Age and period effect





Age and period effect with `ci.exp`

- In rate models there is always one term with the **rate** dimension — usually **age**
- But it must refer to a specific **reference** value for all **other** variables (`P`).
- **All** parameters must be used in computing rates, at some reference value(s).
- For the "other" variables, report the RR **relative** to the reference point.
- Only parameters relevant for the variable (`P`) used.
- Contrast matrix is a **difference** between (splines at) the prediction points and the reference point.

Representation of follow-up data

Epidemiology with R
23 January 2015
Université Bordeaux
http://BendixCarstensen.com/Epi

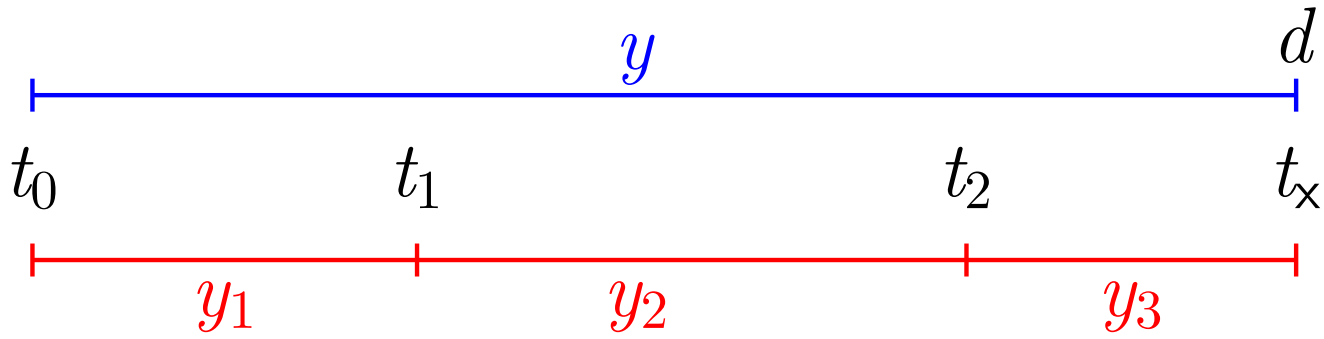
Follow-up studies and rates

- Response **variable** is the rate:
  the **observation** of it is bivariate:
  - D — events, deaths
  - Y — person-years
- Rates differ between persons.
- Rates differ within persons:
  - By age
  - By calendar time
  - By disease duration
  - ...
- Multiple timescales.
- Multiple states (little boxes — later)

Representation of follow-up data

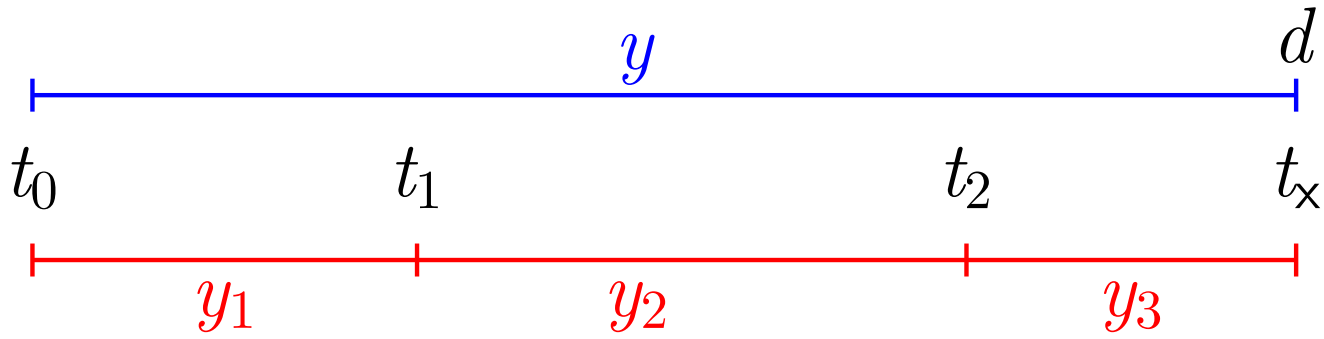
Follow-up **data** for each individual therefore normally have three variables:

Date of entry	`entry`	date variable
Date of exit	`exit`	date variable
Status at exit	`fail`	indicator (0/1)



Probability	log-Likelihood
$\mathrm{P}(d \text{ at } t_x \mid \text{entry } t_0)$	$d\log(\lambda) - \lambda y$
$= \mathrm{P}(\text{surv } t_0 \to t_1 \mid \text{entry } t_0)$	$= 0\log(\lambda) - \lambda y_1$
$\times \mathrm{P}(\text{surv } t_1 \to t_2 \mid \text{entry } t_1)$	$+ 0\log(\lambda) - \lambda y_2$
$\times \mathrm{P}(d \text{ at } t_x \mid \text{entry } t_2)$	$+ d\log(\lambda) - \lambda y_3$

In the constant intensity model, $(\sum d_i, \sum y_i)$ is the sufficient statistic



Probability

log-Likelihood

$\mathrm{P}(d \text{ at } t_x | \text{entry } t_0)$

$$= \mathrm{P}(\text{surv } t_0 \to t_1 | \text{entry } t_0) \qquad = 0 \log(\lambda_1) - \lambda_1 y_1$$
$$\times \mathrm{P}(\text{surv } t_1 \to t_2 | \text{entry } t_1) \qquad + 0 \log(\lambda_2) - \lambda_2 y_2$$
$$\times \mathrm{P}(d \text{ at } t_x | \text{entry } t_2) \qquad + d \log(\lambda_3) - \lambda_3 y_3$$

Dividing follow-up in intervals allows for models with **time-varying** intensity

Dividing follow-up into intervals

If we want to put y and d into intervals on some timescale we must know:

Origin: The date where the time scale is 0:

- Age — 0 at date of birth
- Disease duration — 0 at date of diagnosis
- Occupational exposure — 0 at date of hire

Intervals: How should it be subdivided:

- 1-year classes? 5-year classes?
- Equal length?

Aim: Separate rate in each interval

Timescales

- A timescale is a variable that varies **deterministically** *within* each person during follow-up:
  - Age
  - Calendar time
  - Time since treatment
  - Time since relapse
- All timescales advance at the same pace (1 year per year . . .)
- Note: Cumulative exposure is **not** a timescale.

Follow-up on several timescales

- The risk-time is the same on all timescales
- Only need the entry point on each time scale:
  - Age at entry.
  - Date of entry.
  - Time since treatment at entry.
    — if time of treatment is the entry, this is 0 for all.
- Response variable in analysis of rates:

$$(d, y) \qquad (\text{event}, \text{duration})$$

- Covariates in analysis of rates:
  - timescales
  - other (fixed) measurements

Follow-up data in `Epi` — `Lexis` objects

A follow-up study:

```
> round( th, 2 )
    id sex birthdat contrast injecdat volume exitdat ex
1    1   2  1916.61        1  1938.79     22 1976.79
2  640   2  1896.23        1  1945.77     20 1964.37
3 3425   1  1886.97        2  1955.18      0 1956.59
4 4017   2  1936.81        2  1957.61      0 1992.14
...
```

Timescales of interest:

- Age
- Calendar time
- Time since injection

Definition of `Lexis` object

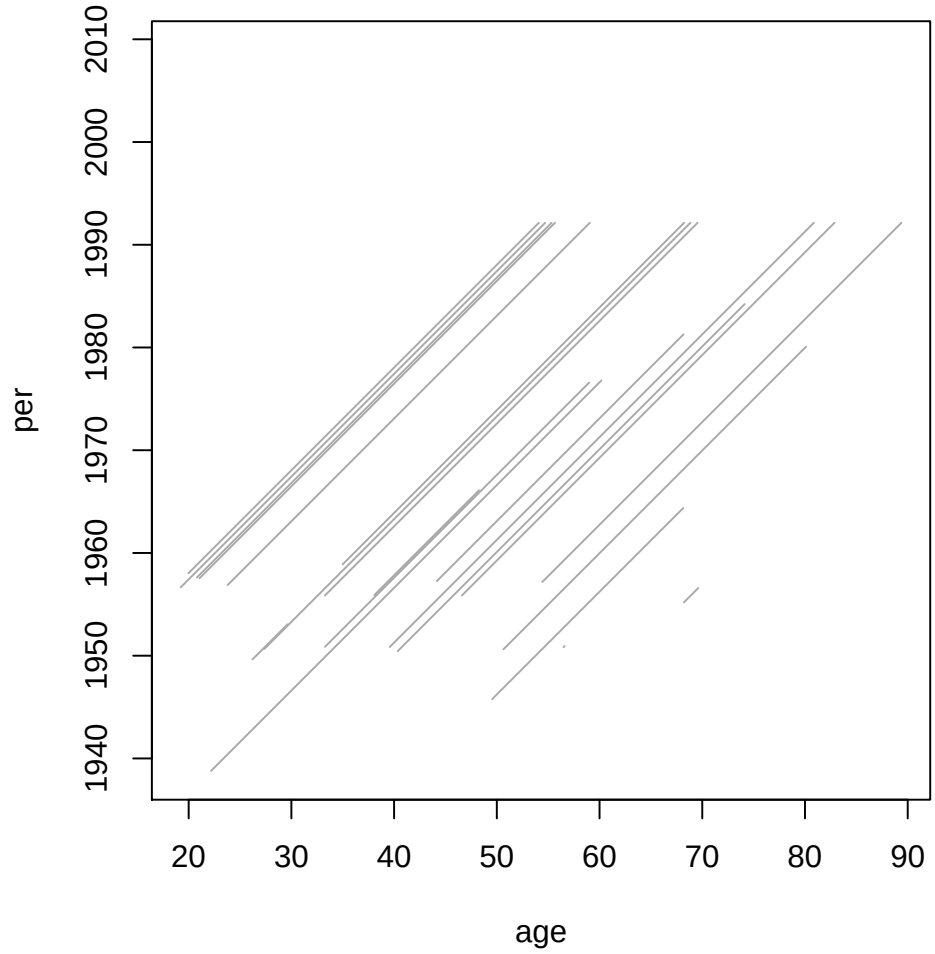
```
> thL <- Lexis( entry = list( age = injecdat-birthdat,
+                             per = injecdat,
+                             tfi = 0 ),
+               exit = list( per = exitdat ),
+         exit.status = as.numeric(exitstat==1),
+               data = th )
```

`entry` is defined on **three** timescales,
but `exit` is only defined on **one** timescale:
Follow-up time is the same on all timescales:

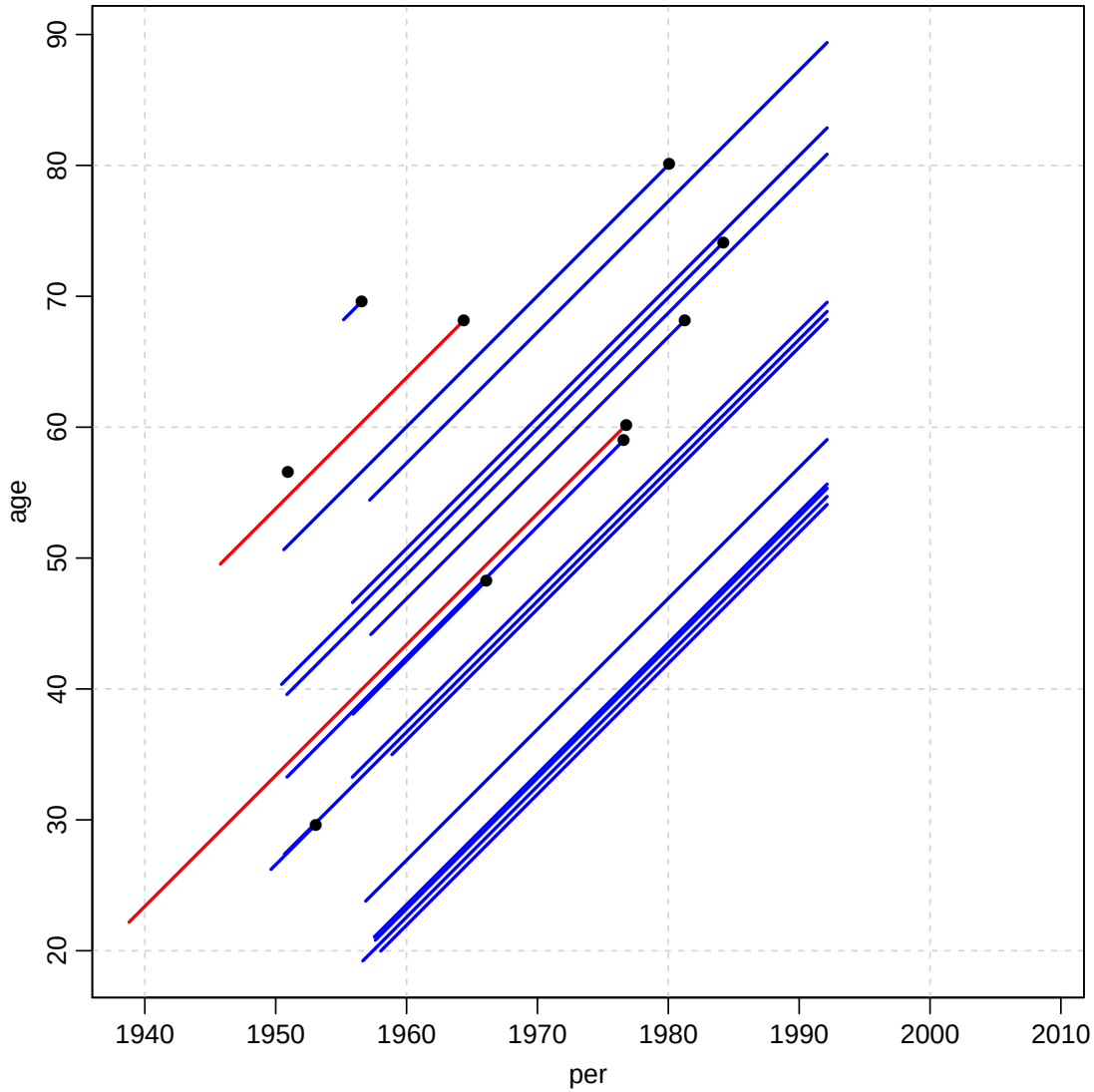
`exitdat - injecdat`

The looks of a `Lexis` object

```
> thL[,1:9]
    age      per tfi lex.dur lex.Cst lex.Xst lex.id
1 22.18 1938.79   0   37.99       0       1      1
2 49.54 1945.77   0   18.59       0       1      2
3 68.20 1955.18   0    1.40       0       1      3
4 20.80 1957.61   0   34.52       0       0      4
...

> summary( thL )
Transitions:
     To
From 0  1 Records:  Events:  Risk time:  Persons:
   0 3 20        23        20      512.59        23
```

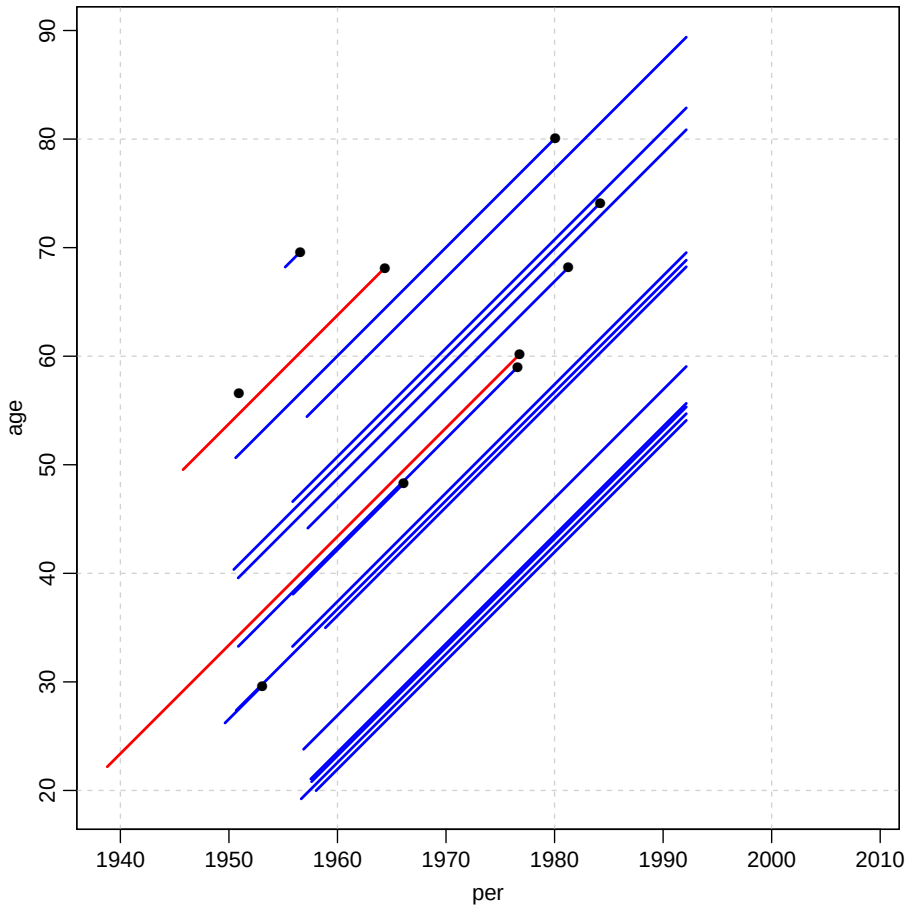

per
1940
1950
1960
1970
1980
1990
2000
2010
20
30
40
50
60
70
80
90
age

```
> plot( thL )
```




```
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> plot( thL, 2:1, lwd=2, col=c("red","blue")[thL$contrast], grid
> points( thL, 2:1, pch=c(NA,16)[thL$lex.Xst+1], lwd=2, cex=1.0
```

Lexis diagram



EINLEITUNG

IN DIE

THEORIE

DER

BEVÖLKERUNGSSTATISTIK

VON

W. LEXIS

DR. DER STAATSWISSENSCHAFTEN UND DER PHILOSOPHIE,
O. PROFESSOR DER STATISTIK IN DORPAT.

STRASSBURG

KARL J. TRÜBNER

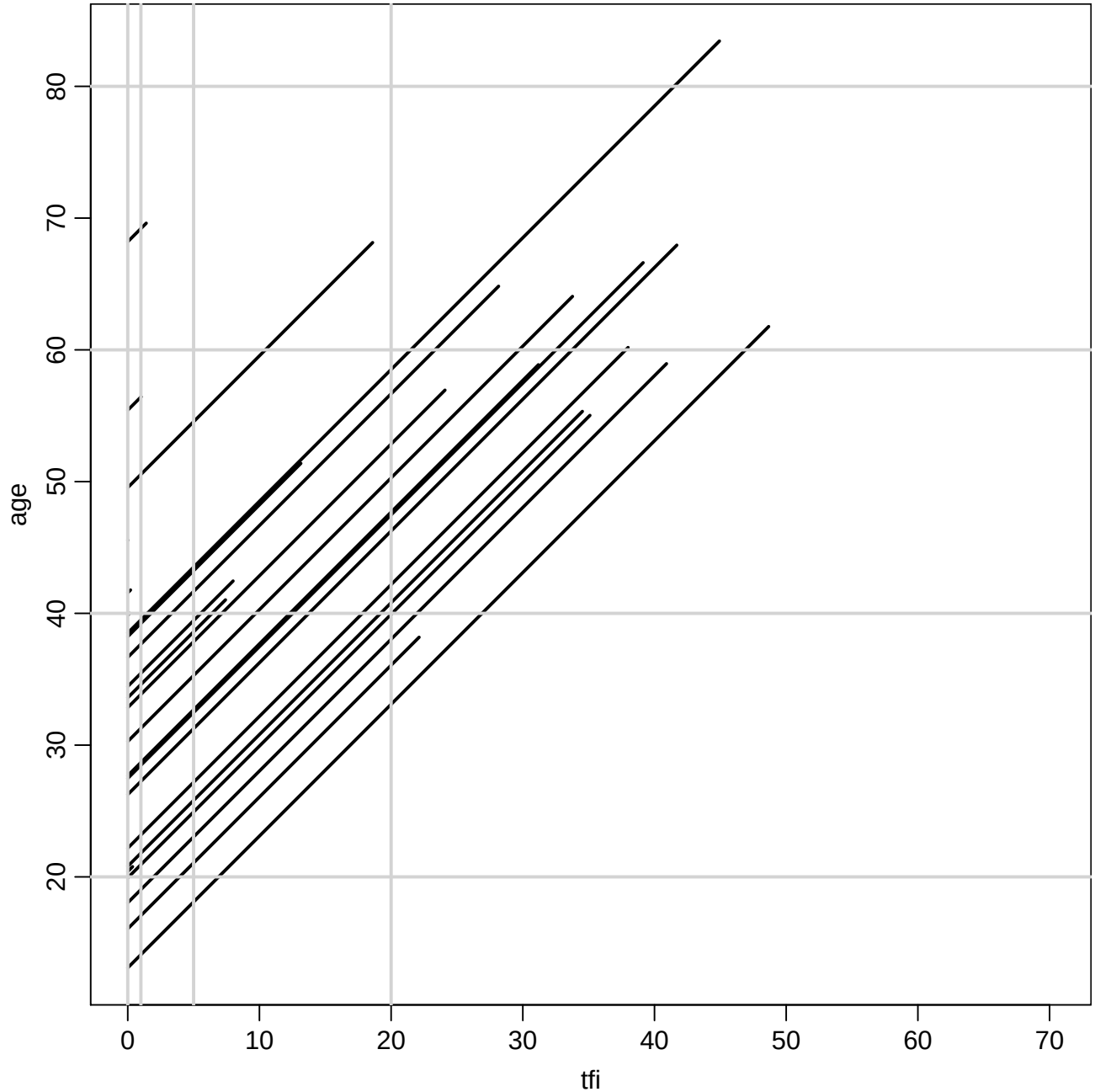
1875.

Splitting follow-up time

```
> spl1 <- splitLexis( thL, breaks=seq(0,100,20),
>                          time.scale="age" )
> round(spl1,1)
   age    per  tfi lex.dur lex.Cst lex.Xst   id sex birthdat con
trast injecdat volume
1 22.2 1938.8  0.0    17.8       0       0    1   2   1916.6
2 40.0 1956.6 17.8    20.0       0       0    1   2   1916.6
3 60.0 1976.6 37.8     0.2       0       1    1   2   1916.6
4 49.5 1945.8  0.0    10.5       0       0  640   2   1896.2
5 60.0 1956.2 10.5     8.1       0       1  640   2   1896.2
6 68.2 1955.2  0.0     1.4       0       1 3425   1   1887.0
7 20.8 1957.6  0.0    19.2       0       0 4017   2   1936.8
8 40.0 1976.8 19.2    15.3       0       0 4017   2   1936.8
...
```

Split on another timescale

```
> spl2 <- splitLexis( spl1, time.scale="tfi",
                                  breaks=c(0,1,5,20,100) )
> round( spl2, 1 )
   lex.id  age    per  tfi lex.dur lex.Cst lex.Xst   id sex birt
dat contrast injecdat volume
1       1 22.2 1938.8  0.0     1.0       0       0    1   2   19
2       1 23.2 1939.8  1.0     4.0       0       0    1   2   19
3       1 27.2 1943.8  5.0    12.8       0       0    1   2   19
4       1 40.0 1956.6 17.8     2.2       0       0    1   2   19
5       1 42.2 1958.8 20.0    17.8       0       0    1   2   19
6       1 60.0 1976.6 37.8     0.2       0       1    1   2   19
7       2 49.5 1945.8  0.0     1.0       0       0  640   2   18
8       2 50.5 1946.8  1.0     4.0       0       0  640   2   18
9       2 54.5 1950.8  5.0     5.5       0       0  640   2   18
10      2 60.0 1956.2 10.5     8.1       0       1  640   2   18
11      3 68.2 1955.2  0.0     1.0       0       0 3425   1   18
12      3 69.2 1956.2  1.0     0.4       0       1 3425   1   18
13      4 20.8 1957.6  0.0     1.0       0       0 4017   2   19
14      4 21.8 1958.6  1.0     4.0       0       0 4017   2   19
15      4 25.8 1962.6  5.0    14.2       0       0 4017   2   19
16      4 40.0 1976.8 19.2     0.8       0       0 4017   2   19
17      4 40.8 1977.6 20.0    14.5       0       0 4017   2   19
...
```



age	tfi	lex.dur
22.2	0.0	1.0
23.2	1.0	4.0
27.2	5.0	12.8
40.0	17.8	2.2
42.2	20.0	17.8
60.0	37.8	0.2

```
plot( spl2, c(1,3), col="black", lwd=2 )
```

Rates constant in small intervals

- This setup is for a situation where it is assumed that rates are constant in each of the intervals.
- Each observation in the dataset contributes a term to a "Poisson" likelihood.
- Rates can vary along several timescales simultaneously.
- Models can include fixed covariates, as well as the timescales (the left end-points of the intervals) as continuous variables.

Practical analysis of time-split data

- d_{pi} — events in the variable `lex.Xst`
  Enters the model as response:
  - `lex.Xst==1`
  - — in general: `lex.Xst=="Event"`
- y_{pi} — risk time: `lex.dur` (duration):
  In the model as offset: `log(lex.dur)`.
- Covariates are:
  - timescales (age, period, time in study)
  - other variables for this person (constant or *assumed* constant in each interval).
- Model rates using the covariates in `glm`:
  — no difference between time-scales and other covariates.

Cox model as the limit of a Poisson model

Epidemiology with R
23 January 2015
Université Bordeaux
http://BendixCarstensen.com/Epi

Cox-Pois

Cox proportional hazards model

$$\lambda_i(t) = \lambda_0(t)\exp\left(\beta_1 x_{1i} + \beta_2 x_{2i} + \ldots\right)$$

- Most common model applied to time-to-event outcomes.
- Makes no assumptions about the **shape** of the underlying hazard function
- However, it does make the assumption that the hazard rates for patient subgroups are proportional over time
- The Cox model models the hazard function, $\lambda_i(t; x_i)$ where x_i denotes the covariate vector

Cox-likelihood

Partial likelihood for regression parameters:

$$\ell(\beta) = \sum_{\text{death times}} \log\left(\frac{\mathrm{e}^{\eta_{\text{death}}}}{\sum_{i\in\mathcal{R}_t}\mathrm{e}^{\eta_i}}\right)$$

is also a **profile likelihood** in the model where observation time has been subdivided in small pieces (empirical rates) and each small piece provided with its own parameter:

$$\log\big(\lambda(t,x)\big) = \log\big(\lambda_0(t)\big) + x'\beta = \alpha_t + \eta$$

Log-likelihood contributions that contain information on a specific time-scale parameter α_t will be from empirical rates (d, y) (wlog assume $y = 1$ for all):

- $(0, 1)$ from those at risk at time t.
- $(1, 1)$ from the death at time t

Note: One contribution from each person at risk to this part of the log-likelihood:

$$\begin{aligned} \ell_t(\alpha_t, \beta) &= \sum_{i \in \mathcal{R}_t} \left\{ d_i(\alpha_t + \eta_i) - \mathrm{e}^{\alpha_t + \eta_i} \right\} \\ &= \alpha_t + \eta_{\text{death}} - \mathrm{e}^{\alpha_t} \sum_{i \in \mathcal{R}_t} \mathrm{e}^{\eta_i} \end{aligned}$$

where η_{death} is the linear predictor for the person that died.

The derivative w.r.t. α_t is:

$$\mathrm{D}_{\alpha_t}\ell(\alpha_t,\beta) = 1 - \mathrm{e}^{\alpha_t}\sum_{i\in\mathcal{R}_t}\mathrm{e}^{\eta_i} = 0 \Leftrightarrow \mathrm{e}^{\alpha_t} = \frac{1}{\sum_{i\in\mathcal{R}_t}\mathrm{e}^{\eta_i}}$$

Feed this back into the log-likelihood for α_t, and get the **profile likelihood** (with α_t "profiled out"):

$$\log\left(\frac{1}{\sum_{i\in\mathcal{R}_t}\mathrm{e}^{\eta_i}}\right) + \eta_{\mathsf{death}} - 1 = \log\left(\frac{\mathrm{e}^{\eta_{\mathsf{death}}}}{\sum_{i\in\mathcal{R}_t}\mathrm{e}^{\eta_i}}\right) - 1$$

— the same as the contribution from time t to Cox's partial likelihood.

What the Cox-model really is

Taking the life-table approach **ad absurdum** by:

- dividing time with max 1 event per interval
- modelling one covariate, the time-scale, with one parameter per time-interval
- profiling these parameters out and maximizing the profile likelihood

Subsequently recover the effect of the timescale by smoothing an estimate of the parameters that was profiled out!

Smooth functions of time

- Not necessary to estimate a separate parameter for each time interval.
- Use a parametric smoother, *e.g.*:
  - fractional polynomials
  - restricted cubic splines (a.k.a. natural splines)
- Fit time dependent effects by fitting an interaction between a covariate and the smooth function
- Provides a direct likelihood ratio test of proportionality (test forinteracton).

The baseline hazard and survival function

A parametric function, h, for the (log) baseline hazard gives the possibility to plot this with confidence intervals for a given set of covariate values, x_0.

The survival function in a multiplicative Poisson model has the form:

$$S(t) = \exp\Big(-\sum_{\tau<t} \exp(h(\tau) + x_0'\beta)\Big)$$

This is just a non-linear function of the parameters in the model, i.e β and the parameters in h. So the variance can be computed using the δ-method.

δ-method for survival function

1. Select timepoints t_i (fairly close).
2. Estimates of log-rates, $f(t_i)$, for $x = x_0$:

$$\hat{f}(t_i) = \mathbf{B}(x_0)\,\hat{\gamma} = \mathbf{B}\,\hat{\gamma}$$

   where γ is the total parameter vector in the model.
3. Variance-covariance matrix of $\hat{\gamma}$: $\hat{\Sigma}$.
4. Variance-covariance of $\hat{f}(t_i)$: $\mathbf{B}\hat{\Sigma}\mathbf{B}'$.
5. Transformation to the rates is the coordinate-wise exponential function, with derivative $\mathrm{diag}\left[\exp\left(\hat{f}(t_i)\right)\right]$
6. Variance-covariance matrix of the rates at the points t_i:

$$\mathrm{diag}(\mathrm{e}^{\hat{f}(t_i)})\,\mathbf{B}\,\hat{\Sigma}\,\mathbf{B}'\,\mathrm{diag}(\mathrm{e}^{\hat{f}(t_i)})'$$

7. Transformation to cumulative hazard (ℓ is interval length):

$$\ell \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} e^{\hat{f}(t_1))} \\ e^{\hat{f}(t_2))} \\ e^{\hat{f}(t_3))} \\ e^{\hat{f}(t_4))} \\ e^{\hat{f}(t_5))} \end{bmatrix} = \mathbf{L} \begin{bmatrix} e^{\hat{f}(t_1))} \\ e^{\hat{f}(t_2))} \\ e^{\hat{f}(t_3))} \\ e^{\hat{f}(t_4))} \\ e^{\hat{f}(t_5))} \end{bmatrix}$$

8. Variance-covariance matrix for the cumulative hazard is:

$$\mathbf{L}\,\mathrm{diag}(e^{\hat{f}(t_i)})\,\mathbf{B}\,\hat{\Sigma}\,\mathbf{B}'\,\mathrm{diag}(e^{\hat{f}(t_i)})'\,\mathbf{L}'$$

9. Construct CIs for cumulative hazard.

10. Transform to survival curve with CIs.

A worked example using Lexis

```
> library( Epi )
> library( splines )
> library( survival )
> data( lung )
> lung[1:5,]
```

	inst	time	status	age	sex	ph.ecog	ph.karno	pat.karno	meal.ca
1	3	306	2	74	1	1	90	100	117
2	3	455	2	68	1	0	90	90	122
3	3	1010	1	56	1	0	90	90	N
4	5	210	2	57	1	1	90	60	115
5	1	883	2	60	1	0	100	90	N

```
> lung$sex <- factor( lung$sex, labels=c("M","F") )
> system.time(
+ c.as <- coxph( Surv( time, status==2 ) ~ age + sex,
+                 method="breslow", eps=10^-8, iter.max=25, data
+             )

   user  system elapsed
  0.008   0.000   0.008
```

```
> summary( c.as )

Call:
coxph(formula = Surv(time, status == 2) ~ age + sex, data = l
    method = "breslow", eps = 10^-8, iter.max = 25)

  n= 228, number of events= 165

           coef exp(coef)  se(coef)      z Pr(>|z|)
age   0.017013  1.017158  0.009222  1.845  0.06506
sexF -0.512565  0.598957  0.167462 -3.061  0.00221

     exp(coef) exp(-coef) lower .95 upper .95
age      1.017     0.9831    0.9989    1.0357
sexF     0.599     1.6696    0.4314    0.8316

Concordance= 0.603  (se = 0.026 )
Rsquare= 0.06   (max possible= 0.999 )
Likelihood ratio test= 14.08  on 2 df,   p=0.0008741
Wald test            = 13.44  on 2 df,   p=0.001208
Score (logrank) test = 13.69  on 2 df,   p=0.001067
```

```
> Lx <- Lexis( exit = list( tfd=time),
+              exit.status = factor(status,labels=c("Alive","Dea
+              data=lung )

   NOTE: entry.status has been set to "Alive" for all.
   NOTE: entry is assumed to be 0 on the tfd timescale.

> summary( Lx )

   Transitions:
        To
   From    Alive Dead  Records:  Events: Risk time:  Persons:
     Alive    63  165       228      165      69593       228

> dx <- splitLexis( Lx, "tfd", breaks=c(0,unique(Lx$time)) )
> summary( dx )

   Transitions:
        To
   From    Alive Dead  Records:  Events: Risk time:  Persons:
     Alive 19857  165     20022      165      69593       228

> subset( dx, lex.id==19 )[,1:13]
```

	lex.id	tfd	lex.dur	lex.Cst	lex.Xst	inst	time	status	age
2139	19	0	5	Alive	Alive	1	61	2	56
2140	19	5	6	Alive	Alive	1	61	2	56
2141	19	11	1	Alive	Alive	1	61	2	56
2142	19	12	1	Alive	Alive	1	61	2	56
2143	19	13	2	Alive	Alive	1	61	2	56
2144	19	15	11	Alive	Alive	1	61	2	56
2145	19	26	4	Alive	Alive	1	61	2	56
2146	19	30	1	Alive	Alive	1	61	2	56
2147	19	31	22	Alive	Alive	1	61	2	56
2148	19	53	1	Alive	Alive	1	61	2	56
2149	19	54	5	Alive	Alive	1	61	2	56
2150	19	59	1	Alive	Alive	1	61	2	56
2151	19	60	1	Alive	Dead	1	61	2	56

— much larger dataset, but same y and d.

```
> system.time(
+ p.as <- glm( (dx$lex.Xst=="Dead") ~ factor(tfd) + age + sex,
+               offset = log(lex.dur),
+               family=poisson, data=dx, eps=10^-8, maxit=25 ) )

   user  system elapsed
 14.375   0.024  14.393
```

```
> sum( !is.na(coef(p.as)) )

  [1] 188

> round( rbind( ci.lin(p.as,subset=c("age","sex")),
+               ci.lin(c.as) ), 4 )

      Estimate StdErr       z      P    2.5%   97.5%
age     0.0170 0.0092  1.8448 0.0651 -0.0011  0.0351
sexF   -0.5126 0.1675 -3.0608 0.0022 -0.8408 -0.1843
age     0.0170 0.0092  1.8448 0.0651 -0.0011  0.0351
sexF   -0.5126 0.1675 -3.0608 0.0022 -0.8408 -0.1843

> ci.lin(p.as,subset=c("age","sex")) / ci.lin(c.as)

      Estimate StdErr z         P      2.5% 97.5%
age          1      1 1 0.9999999 0.9999997     1
sexF         1      1 1 0.9999998 1.0000000     1
```

— same model:

186 parameters describing the baseline hazard!

```
> kn <- c(0,25,75,150,250,500,1000)
> system.time(
+ s.as <- glm( (lex.Xst=="Dead") ~ Ns( tfd, knots=kn )
+                                 + age + sex,
+              offset = log(lex.dur),
+              family = poisson, data=dx, eps=10^-8, maxit=25 )
+            )

   user  system elapsed
  0.286   0.000   0.286

> length( coef( s.as ) )

[1] 9

> round( rbind( ci.lin(s.as,subset=c("age","sex")),
+               ci.lin(c.as) ), 4 )

      Estimate StdErr       z      P    2.5%   97.5%
age     0.0164 0.0092  1.7783 0.0754 -0.0017  0.0344
sexF   -0.5120 0.1675 -3.0576 0.0022 -0.8402 -0.1838
age     0.0170 0.0092  1.8448 0.0651 -0.0011  0.0351
sexF   -0.5126 0.1675 -3.0608 0.0022 -0.8408 -0.1843
```

```
> round( ci.lin(s.as,subset=c("age","sex")) / ci.lin(c.as), 4 )

      Estimate StdErr      z      P   2.5%  97.5%
age     0.9621 0.9982 0.9639 1.1583 1.5751 0.9807
sexF    0.9989 0.9999 0.9990 1.0107 0.9993 0.9970
```

```
> par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6,
+      las=1, bty="n" )
> # baseline intensity from Poisson model
> new <- data.frame( tfd = 1:100*10,
+                    sex = "M",
+                    age = 60,
+                lex.dur = 10 )
> s.pr <- ci.pred( s.as, newdata=new )
> matplot( new$tfd, s.pr*300, log="y", ylim=c(1,50),
+          xlab="Time since diagnosis (days)",
+          ylab="Mortality (%/month)",
+          type="l", lty=1, lwd=c(4,1,1), col="black" )
> # survival function, Cox model, Breslow estimator
> plot( survfit( c.as, newdata=data.frame(age=60,sex="M") ),
+       conf.int=TRUE, mark.time=FALSE, lwd=1, ylim=0:1, yaxs="i
```

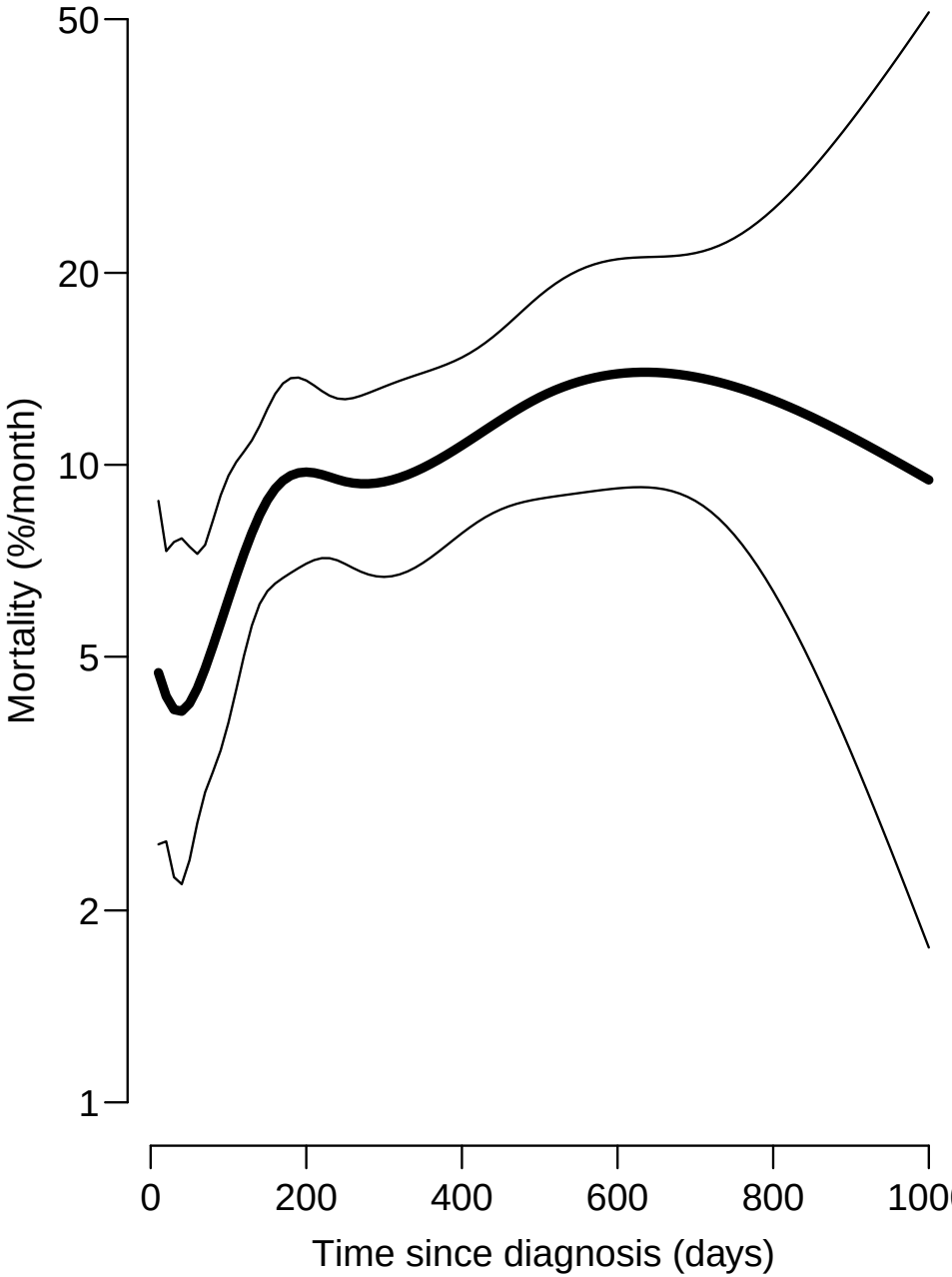
```
+       xlab="Time since diagnosis (days)",
+       ylab="Survival probability")
> lines( survfit( c.as, newdata=data.frame(age=60,sex="M") ),
+       conf.int=FALSE, mark.time=FALSE, lwd=3 )
> # survival function from Poisson model
> round( ci.lin( s.as ), 3 )

                          Estimate StdErr      z     P   2.5%  97.5
(Intercept)                -7.332  0.780 -9.395 0.000 -8.862 -5.80
Ns(tfd, knots = kn)1       -0.012  0.591 -0.021 0.984 -1.170  1.14
Ns(tfd, knots = kn)2        0.730  0.620  1.177 0.239 -0.485  1.94
Ns(tfd, knots = kn)3        0.434  0.591  0.734 0.463 -0.725  1.59
Ns(tfd, knots = kn)4        1.378  0.588  2.342 0.019  0.225  2.53
Ns(tfd, knots = kn)5        0.560  1.284  0.436 0.663 -1.957  3.07
Ns(tfd, knots = kn)6        0.771  0.909  0.849 0.396 -1.010  2.55
age                         0.016  0.009  1.778 0.075 -0.002  0.03
sexF                       -0.512  0.167 -3.058 0.002 -0.840 -0.18

> CM = cbind( 1, Ns( new$tfd, knots=kn ), 60, 0 )
> round( head(CM), 4 )
```

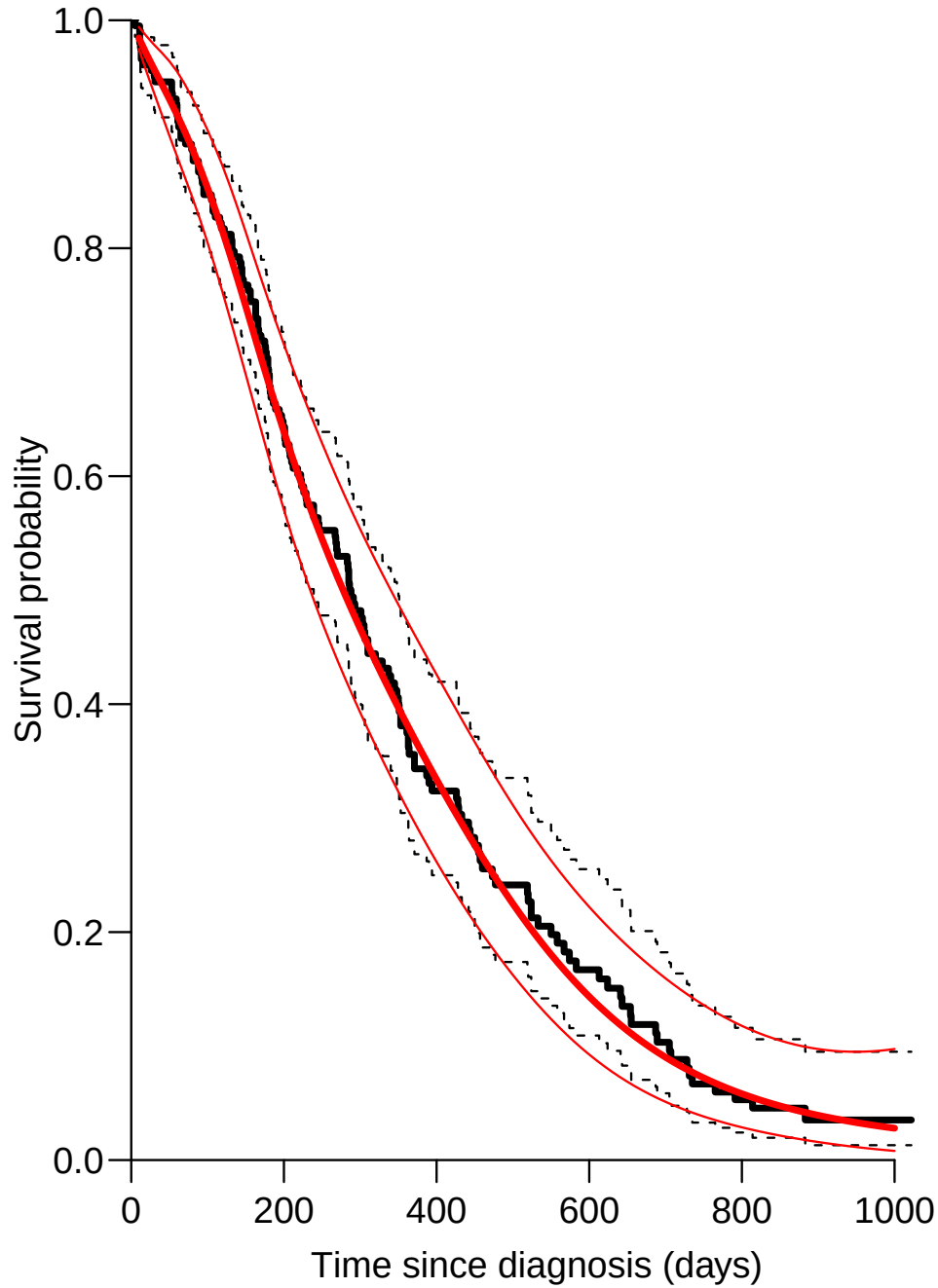
```
             1      2 3       4      5       6
[1,] 1 0.0036 0.0000 0 -0.0916 0.2291 -0.1374 60 0
[2,] 1 0.0284 0.0000 0 -0.1665 0.4162 -0.2497 60 0
[3,] 1 0.0951 0.0001 0 -0.2083 0.5208 -0.3125 60 0
[4,] 1 0.2036 0.0024 0 -0.2139 0.5349 -0.3209 60 0
[5,] 1 0.3333 0.0111 0 -0.1932 0.4831 -0.2898 60 0
[6,] 1 0.4631 0.0305 0 -0.1566 0.3914 -0.2348 60 0
```

```
> head( Lambda <- ci.cum( s.as, ctr.mat=CM, intl=10 ) )
```

```
       Estimate       2.5%      97.5%    Std.err.
[1,] 0.01573572 0.00597744 0.02549399 0.004978803
[2,] 0.03018553 0.01409597 0.04627510 0.008209113
[3,] 0.04395497 0.02175695 0.06615300 0.011325733
[4,] 0.05764164 0.02890228 0.08638099 0.014663206
[5,] 0.07172113 0.03665723 0.10678504 0.017890076
[6,] 0.08658460 0.04604430 0.12712490 0.020684205
```

```
> matlines( new$tfd, exp(-Lambda[,1:3]), lty=1, lwd=c(3,1,1), co
```


50
20
10
5
2
1
Mortality (%/month)
0
200
400
600
800
1000
Time since diagnosis (days)


1.0
0.8
0.6
0.4
0.2
0.0
Survival probability
0
200
400
600
800
1000
Time since diagnosis (days)

Conclusion

- Cox-model only useful for the simplest limited tasks
- Cox model assumptions are counter-intuitive
- Poisson model gives easy access to all aspects of the mortality
  — useful in calculation of probabilities in multistate models
- Poisson model allows "test for proportionality" to be conducted as what it is:
  Test for interaction between time and oter covariates
- Time(scale) is treated as a covariate — trivial to include more timescales

Multiple time-scales: Mortality in Danish DM patients

Epidemiology with R
23 January 2015
Université Bordeaux
http://BendixCarstensen.com/Epi

DM-mort

Mortality among Danish diabetes patients

. . . depends on:

- Age
- Calendar time
- Duration of diabetes

- Age at diagnosis / attained age (current age)
- Calendar time at diagnosis / current date

Timescale or entry time?

```
> LL <- Lexis( entry = list( A = dodm-dobth,
+                            P = dodm,
+                          dur = 0 ),
+              exit = list( P = dox ),
+       exit.status = factor( !is.na(dodth),
+                             labels=c("Alive","Dead") ),
+              data = DMlate )

   NOTE: entry.status has been set to "Alive" for all.

> summary( LL )

   Transitions:
        To
   From    Alive Dead  Records:  Events: Risk time:  Persons:
     Alive  7497 2499       9996     2499   54273.27       9996
```

Three time-scales defined, exit status is death.

Subdivide time by age

```
> SL <- splitLexis( LL, breaks=seq(0,125,1/2), time.scale="A" )
> summary( SL )

   Transitions:
        To
   From     Alive Dead  Records:  Events: Risk time:  Persons:
     Alive 115974 2499    118473     2499   54273.27      9996

> SL[SL$lex.id==80,1:11]

     lex.id        A        P        dur    lex.dur lex.Cst lex
   972     80 65.45654 1997.828 0.00000000 0.04346338   Alive   A
   973     80 65.50000 1997.871 0.04346338 0.50000000   Alive   A
   974     80 66.00000 1998.371 0.54346338 0.50000000   Alive   A
   975     80 66.50000 1998.871 1.04346338 0.50000000   Alive   A
   976     80 67.00000 1999.371 1.54346338 0.50000000   Alive   A
   977     80 67.50000 1999.871 2.04346338 0.11122519   Alive
```

Subdivided time by age

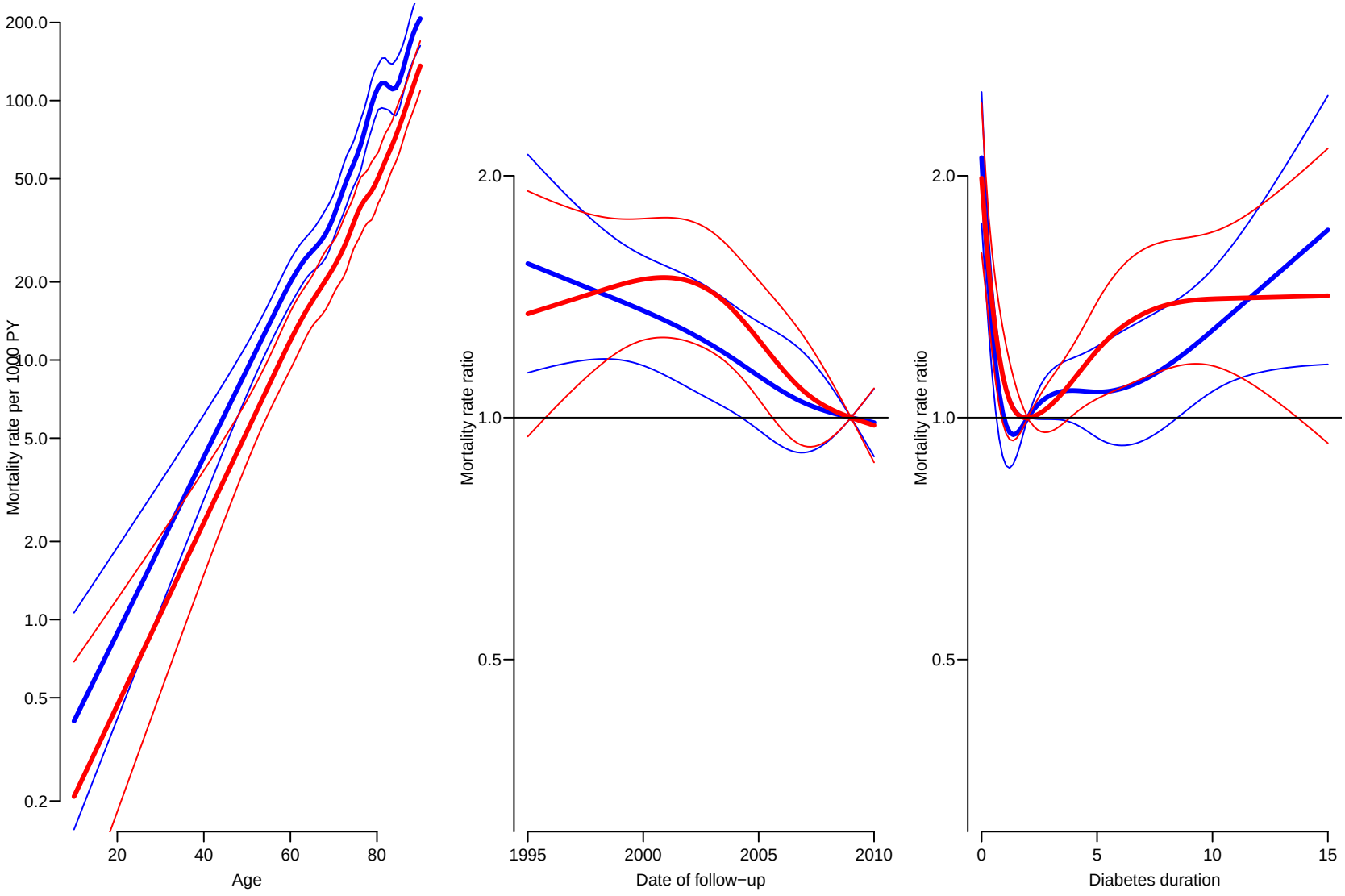
- Small intervals: immaterial which time-scale is used for splitting
- Each interval is defined on all three time-scales
- Time-scales will appear in models a covariates
- . . . normally modelled non-linearly

Sketch of modelling

```
> mm <- glm( (lex.Xst=="Dead") ~ Ns( A, kn=kn.A ) +
+                                Ns( P, kn=kn.P ) +
+                                Ns( dur, kn=kn.dur ),
+            offset = log( lex.dur ),
+            family = poisson,
+              data = subset( SL, sex=="M" ) )
```

- `kn.A`, `kn.P` `kn.dur` predefined vectors of knots for the splines
- `Ns` a function defining a spline basis (columns of the design matrix)
- Use `ci.exp` to extract rates and RRs

```
> m.A <- ci.exp( mm, ctr.mat=cbind(1,AC,PR,dR) ) * 1000
> m.P <- ci.exp( mm, subset="P", ctr.mat=PC-PR )
```

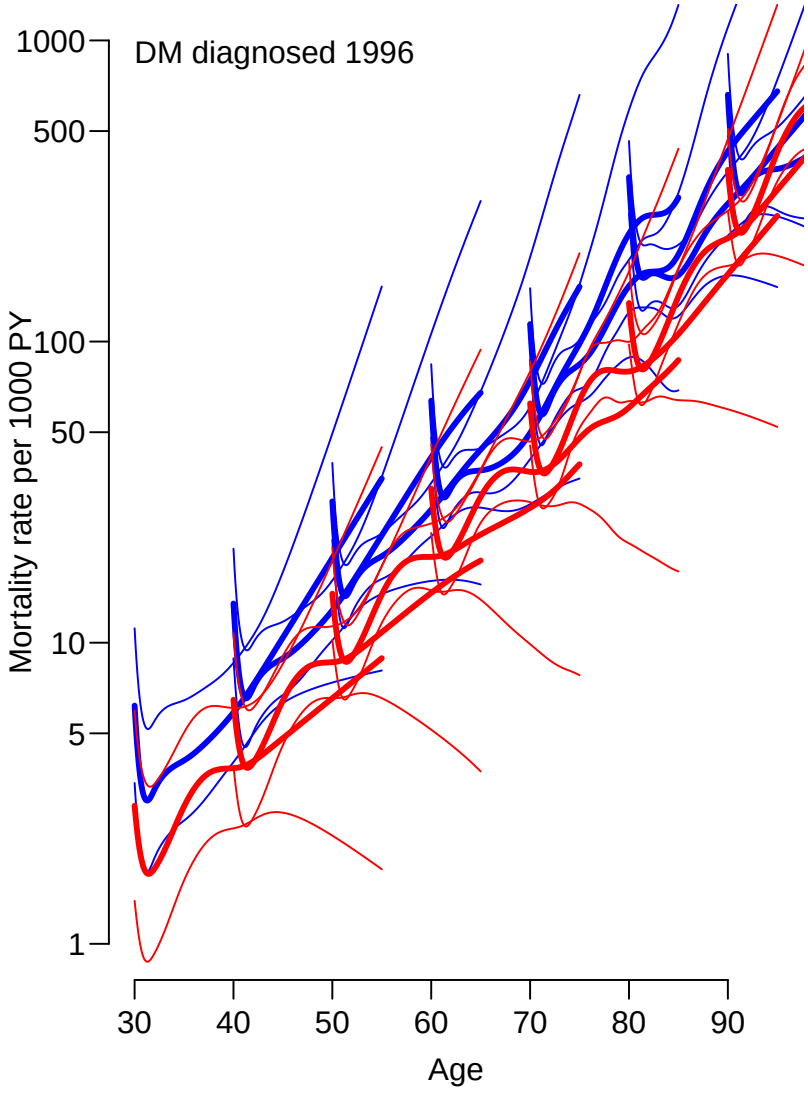


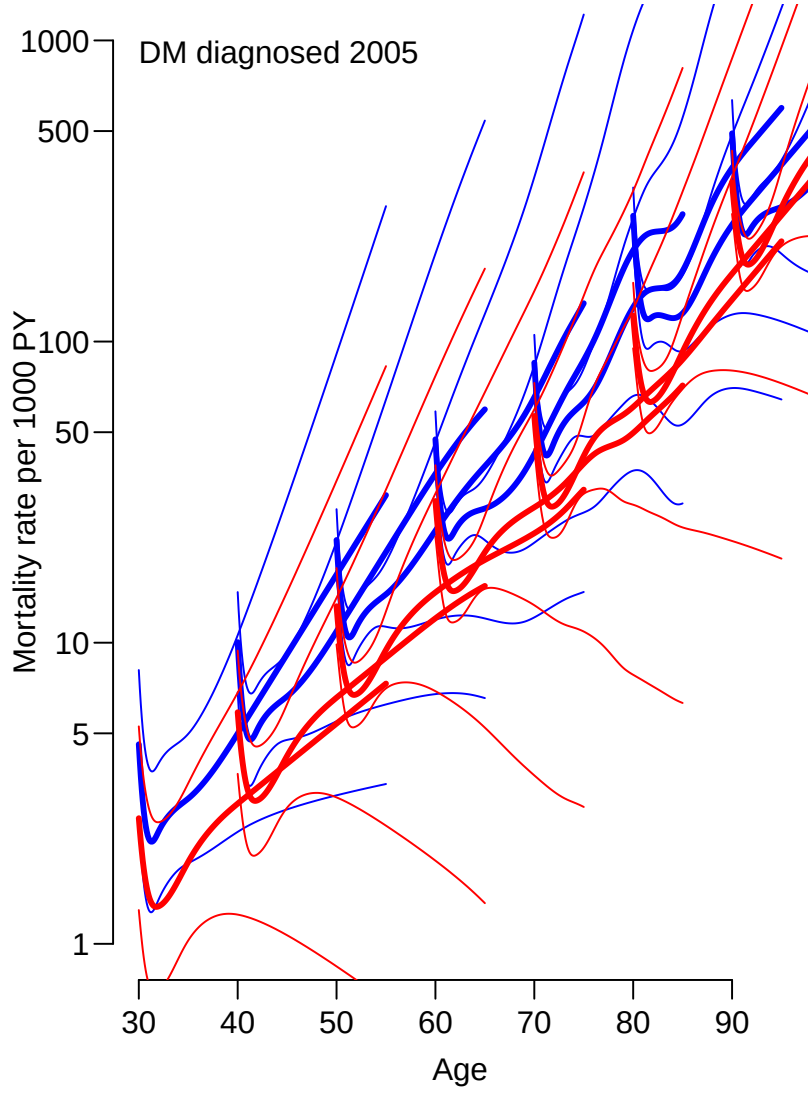
Multiple time-scales:

Mortality in Danish DM patients

Multiple timescales: reporting

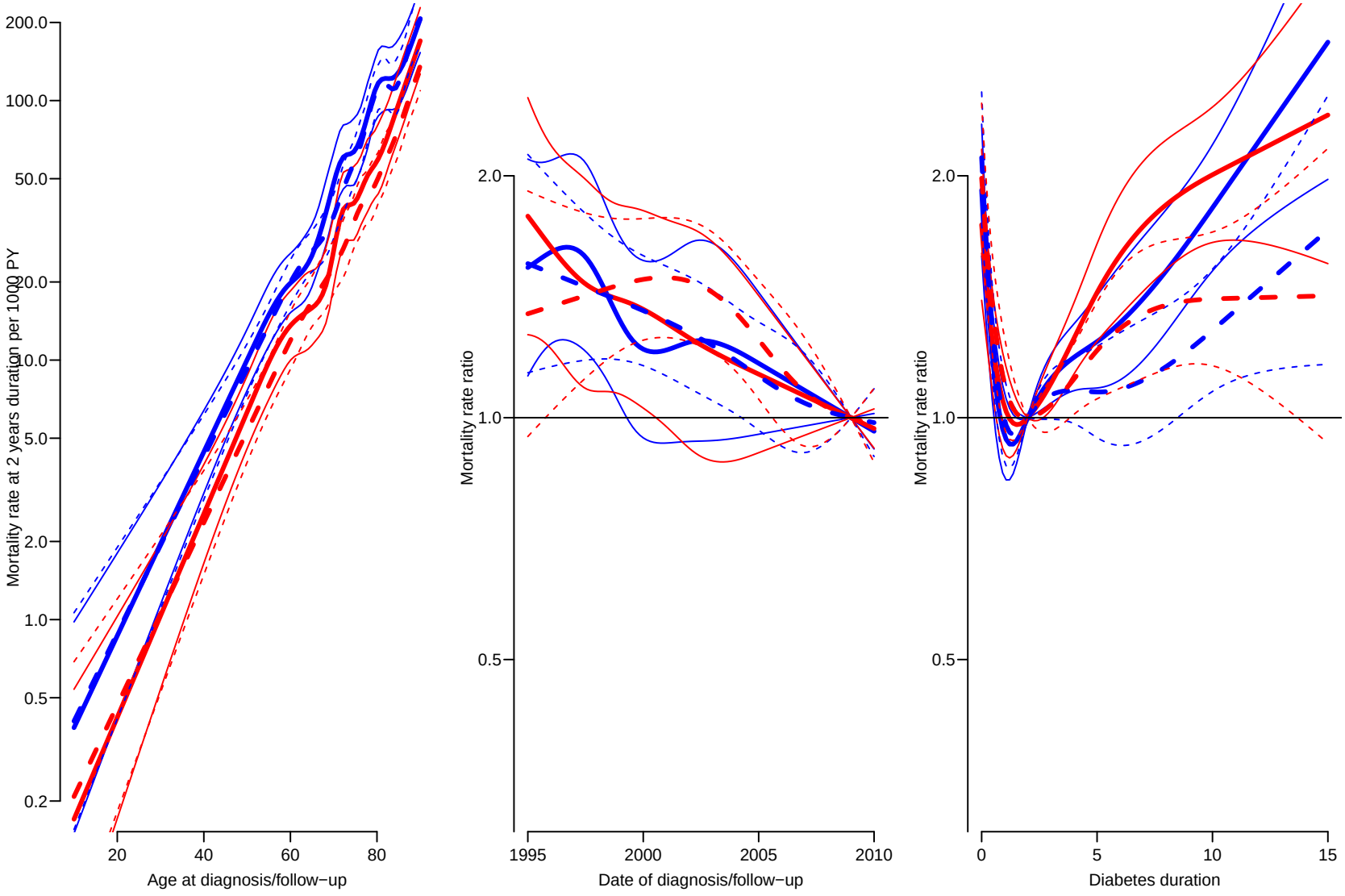
- All timescales advance at the same pace
- Traditional reporting of effects:
  effect of x, assuming all other covariates equal
- Nonsense for multiple time scales
- Joint reporting is needed


DM diagnosed 1996
Mortality rate per 1000 PY
1000
500
100
50
10
5
1
30
40
50
60
70
80
90
Age


DM diagnosed 2005
Mortality rate per 1000 PY
1000
500
100
50
10
5
1
30
40
50
60
70
80
90
Age

Current age or age at diagnosis?

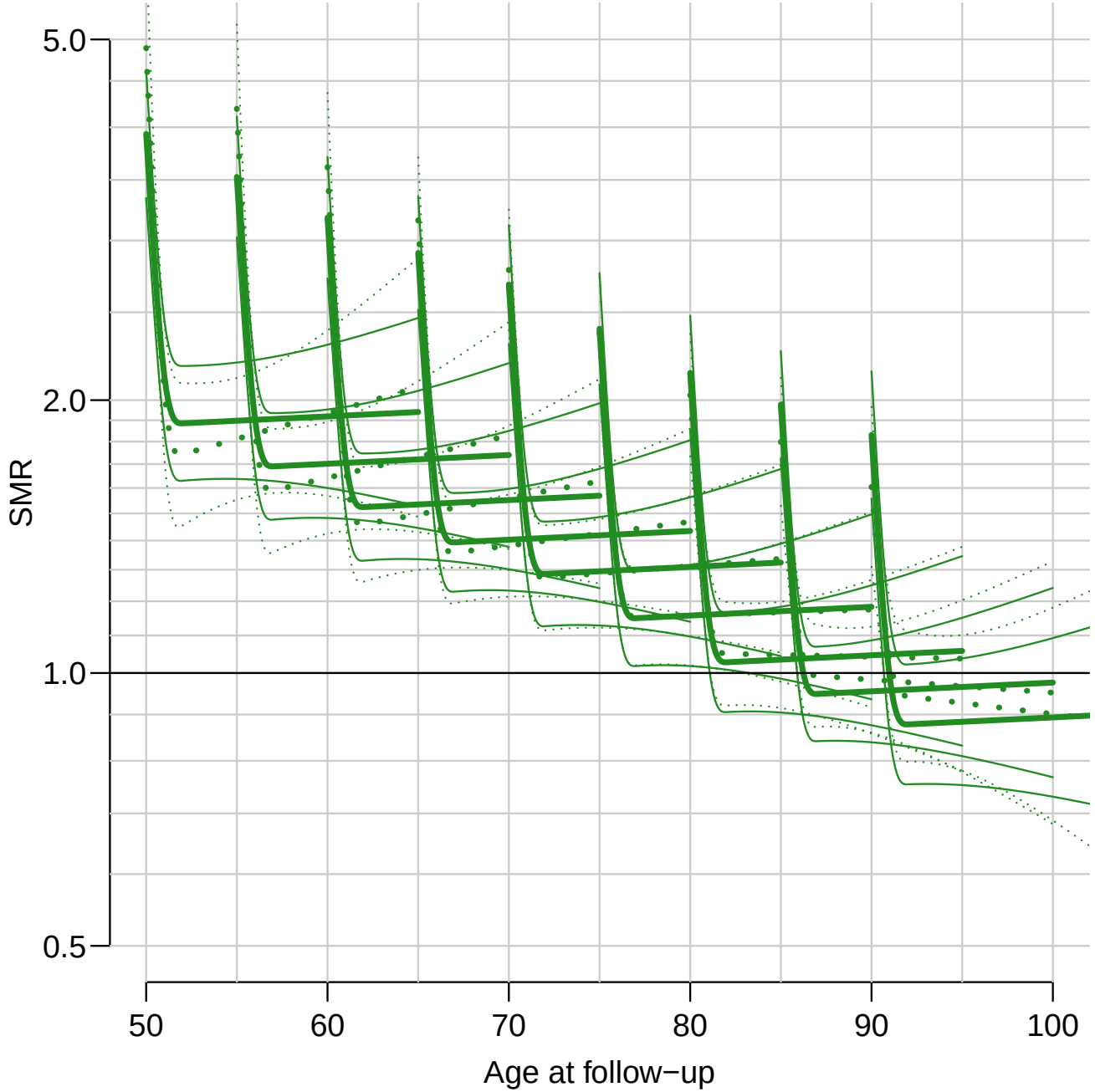
- Not a theoretical question
- It is an empirical question:
- . . . which fits data best
- include both, but:
- current age - age at diagnosis = duration of disease
- Age-Period-Cohort models — separate topic: `http://BendixCarstensen/APC`


Mortality rate at 2 years duration per 1000 PY
200.0
100.0
50.0
20.0
10.0
5.0
2.0
1.0
0.5
0.2
20
40
60
80
Age at diagnosis/follow−up
Mortality rate ratio
2.0
1.0
0.5
1995
2000
2005
2010
Date of diagnosis/follow−up
Mortality rate ratio
2.0
1.0
0.5
0
5
10
15
Diabetes duration

SMR - standardised mortality ratio

- What is the **relative** mortality among DM patienst relative to the entire Danish population?
- Model the same dataset, but using the **expected** numbers of deaths as offset.

```
> SLr <- merge( SL, M.dk[,c("A","P","sex","rate")] )
> SLr$E <- SLr$lex.dur * SLr$rate / 1000
> Sm <- glm( (lex.Xst=="Dead") ~ Ns( A-dur, kn=kn.Ad ) +
+                                Ns( P-dur, kn=kn.Pd ) +
+                                Ns( dur, kn=kn.dur ),
+            offset = log( E ),
+            family = poisson,
+              data = SLr )
```


5.0
2.0
1.0
0.5
SMR
50
60
70
80
90
100
Age at follow−up

Conclusion

- Multiple timescales necessary to describe real data
- Simplest accommodated in Poisson models for time-split data
- Possible, but painful in Cox-type semiparamtric setting
- Reporting models with multiple timescales cannot be done by parts
- Choice of timescale(s) is not a theoretical question — it is an **empirical** question [1, 2, 3]

References

S. Iacobelli and B. Carstensen.
Multiple time scales in multi-state models.
Stat Med, 32(30):5315–5327, Dec 2013.

Martyn Plummer and Bendix Carstensen.
Lexis: An R class for epidemiological studies with long-term follow-up.
Journal of Statistical Software, 38(5):1–12, 1 2011.

Bendix Carstensen and Martyn Plummer.
Using Lexis objects for multi-state models in R.
Journal of Statistical Software, 38(6):1–18, 1 2011.