

# R implementation of the linear ERR model: applications to radiation epidemiology

D. Moriña, P. Gradowska, M. Hauptmann, E. Cardis

October 27th 2015, CREAL



# Contents

- ① Introduction
- ② The package `fit.linERR`
- ③ Examples
- ④ Practical application

## The linear excess relative risk model

- Analysis of cohort and case control data often follow from risk-set sampling designs
- Risk-set sampling designs are related to the Cox proportional hazards model
- Standard statistical packages facilities for fitting Cox or conditional logistic regression models are limited to log-linear models of the form

$$\log(\phi(z, \beta)) = z\beta,$$

where  $z$  is a vector of explanatory variables and  $\phi$  is the rate ratio

- This implies exponential dose-response trends and multiplicative interactions

## The linear excess relative risk model

- Some biological mechanisms are better represented by other model forms
- They may also facilitate exposure-response analyses
- The linear excess relative risk (ERR) model

$$\phi(d, \beta) = 1 + \beta \cdot d$$

is one model form of interest, particularly in environmental and occupational epidemiology

- The R package `fit.linERR` implements a broad class of relative risk models mixing linear and log-linear terms

$$\phi(z_1, \dots, z_k, d, \alpha_1, \dots, \alpha_k, \beta) = e^{\alpha_1 \cdot z_1 + \dots + \alpha_k \cdot z_k} (1 + \beta \cdot d)$$

## The linear excess relative risk model

- Some biological mechanisms are better represented by other model forms
- They may also facilitate exposure-response analyses
- The linear excess relative risk (ERR) model

$$\phi(d, \beta) = 1 + \beta \cdot d$$

is one model form of interest, particularly in environmental and occupational epidemiology

- The R package `fit.linERR` implements a broad class of relative risk models mixing linear and log-linear terms

$$\phi(z_1, \dots, z_k, d, \alpha_1, \dots, \alpha_k, \beta) = e^{\alpha_1 \cdot z_1 + \dots + \alpha_k \cdot z_k} (1 + \beta \cdot d)$$

## The linear excess relative risk model

- The likelihood contribution of each risk set can be written as

$$L(\alpha_1, \dots, \alpha_k, \beta) = \frac{\phi_{case}(z, d, \alpha_1, \dots, \alpha_k, \beta)}{\phi(z, d, \alpha_1, \dots, \alpha_k, \beta)}$$

## The linear excess relative risk model

- Despite its advantages, the linear ERR model has worst statistical properties than the standard Cox model
- Parameter estimates may be inaccurate if the number of events is not large
- Large number of events may result in large computational time
- Parameter estimates may be under the lower boundary of the feasible region  $\left(\frac{-1}{D_{max}}\right)$
- In such cases no estimate of the standard error can be obtained

## The linear excess relative risk model

- Despite its advantages, the linear ERR model has worst statistical properties than the standard Cox model
- Parameter estimates may be inaccurate if the number of events is not large
- Large number of events may result in large computational time
- Parameter estimates may be under the lower boundary of the feasible region  $\left(\frac{-1}{D_{max}}\right)$
- In such cases no estimate of the standard error can be obtained



## The linear excess relative risk model

- Despite its advantages, the linear ERR model has worst statistical properties than the standard Cox model
- Parameter estimates may be inaccurate if the number of events is not large
- Large number of events may result in large computational time
- Parameter estimates may be under the lower boundary of the feasible region  $\left(\frac{-1}{D_{max}}\right)$
- In such cases no estimate of the standard error can be obtained

## The linear excess relative risk model

- Despite its advantages, the linear ERR model has worst statistical properties than the standard Cox model
- Parameter estimates may be inaccurate if the number of events is not large
- Large number of events may result in large computational time
- Parameter estimates may be under the lower boundary of the feasible region  $\left(\frac{-1}{D_{max}}\right)$ 
  - In such cases no estimate of the standard error can be obtained

## The linear excess relative risk model

- Despite its advantages, the linear ERR model has worst statistical properties than the standard Cox model
- Parameter estimates may be inaccurate if the number of events is not large
- Large number of events may result in large computational time
- Parameter estimates may be under the lower boundary of the feasible region ( $\frac{-1}{D_{max}}$ )
- In such cases no estimate of the standard error can be obtained

## The package `fit.linERR`

- The likelihood function is written in **C** to improve computing times

## The package `fit.linERR`

- The likelihood function is written in **C** to improve computing times
- The main function returns an object that can be summarized via `summary` command

## The package `fit.linERR`

- The likelihood function is written in **C** to improve computing times
- The main function returns an object that can be summarized via `summary` command
- The printed summary is very similar to that of `Epicure`

Parameter Summary Table:

	Estimate	Std. Error	Test Stat.	p-value
dose	0.10764506	0.086609701	1.2428753	2.139138e-01
ex2\$ds	4.35150398	0.281627097	15.4512972	7.394023e-54
ex2\$ds2	0.04891001	0.245572672	0.1991671	8.421320e-01
ex2\$ds3	-0.00212656	0.008645206	-0.2459814	8.056966e-01

AIC: 1291.526

Deviance: 1283.526

## The package `fit.linERR`

- The likelihood function is written in **C** to improve computing times
- The main function returns an object that can be summarized via `summary` command
- The printed summary is very similar to that of `Epicure`

Parameter Summary Table					
#	Name	Estimate	Std.Err.	Test Stat.	P value
Log-linear term 0					
1	ds.....	4.352	0.2817	15.45	< 0.001
2	ds2.....	0.04940	0.2456	0.2011	> 0.5
3	ds3.....	-0.002126	0.008645	-0.2459	> 0.5
Linear term 1					
4	cum.....	0.1077	0.09218	1.168	0.243
Records used		150000			
-2 * log-likelihood		1283.526	Free parameters		4
AIC		1291.526	Informative risk sets		84

## The package `fit.linERR`

### `fit.linERR`

The main function needs the following arguments:

- `data`: Data frame containing the cohort
- `beta`: Starting values for parameter estimates. Its default value is `NULL`
- `id`: Subject identification
- `max.exp`: Number of exposures
- `status`: Case / control status
- `start`: Age at start of follow-up
- `end`: Age at end of follow-up
- `ages`: Age at each exposure
- `doses`: Dose at each exposure
- `covariates1`: Covariates on the log-linear term. Its default value is `NULL`
- `covariates2`: Covariates on the linear term. Its default value is `NULL`
- `strata`: Strata variable. Its default value is `NULL`



# The package `fit.linERR`

## Data structure

Essentially, the function `fit.linERR` needs a `data.frame` with age and dose at each exposure:

```
> head(ex1)
  id sex entryyear  entryage  exitage  leu   ctage1 ctage2 ctage3 ctage4 ctage5 ctage6 ctage7 ctage8
1  1  1    2007  16.72100  19.7210  0  16.72100    0    0    0    0    0    0    0
2  2  2    1998   5.64140  17.6410  0   5.64140    0    0    0    0    0    0    0
3  3  3    1994  3.93900  19.9390  0   3.93900    0    0    0    0    0    0    0
4  4  4    2009   4.06670   5.0667  0   4.06670    0    0    0    0    0    0    0
5  5  5    1987   0.76266  23.7630  0   0.76266    0    0    0    0    0    0    0
6  6  6    1998   2.77630  14.7760  0   2.77630    0    0    0    0    0    0    0

  ctage9 ctage10 ctage11 ctage12 ctage13 ctage14 ctage15 ctage16 ctage17 ctage18 ctage19 ctage20
1      0      0  0.000      0      0      0      0      0      0      0  0.000      0
2      0      0 15.041      0      0      0      0      0      0      0  0.000      0
3      0      0  0.000      0      0      0      0      0      0      0  0.000      0
4      0      0  0.000      0      0      0      0      0      0      0  0.000      0
5      0      0  0.000      0      0      0      0      0      0      0 18.163      0
6      0      0  0.000      0      0      0      0      0      0      0  0.000      0

  ctage21 ctage22 ctage23 ctage24 ctage25 ctage26 ctage27 ctage28 ctage29 ctage30 ctage31 ctage32
1      0      0      0      0      0      0      0      0      0      0      0      0
2      0      0      0      0      0      0      0      0      0      0      0      0
3      0      0      0      0      0      0      0      0      0      0      0      0
4      0      0      0      0      0      0      0      0      0      0      0      0
5      0      0      0      0      0      0      0      0      0      0      0      0
6      0      0      0      0      0      0      0      0      0      0      0      0

...

```

# The package `fit.linERR`

## Data structure

...

	dose1	dose2	dose3	dose4	dose5	dose6	dose7	dose8	dose9	dose10	dose11	dose12	dose13	dose14	dose15
1	2.8	0	0	0	0	0	0	0	0	0	0.0	0	0	0	0
2	13.0	0	0	0	0	0	0	0	0	0	4.4	0	0	0	0
3	15.0	0	0	0	0	0	0	0	0	0	0.0	0	0	0	0
4	8.6	0	0	0	0	0	0	0	0	0	0.0	0	0	0	0
5	16.3	0	0	0	0	0	0	0	0	0	0.0	0	0	0	0
6	16.1	0	0	0	0	0	0	0	0	0	0.0	0	0	0	0

	dose16	dose17	dose18	dose19	dose20	dose21	dose22	dose23	dose24	dose25	dose26	dose27	dose28
1	0	0	0	0.0	0	0	0	0	0	0	0	0	0
2	0	0	0	0.0	0	0	0	0	0	0	0	0	0
3	0	0	0	0.0	0	0	0	0	0	0	0	0	0
4	0	0	0	0.0	0	0	0	0	0	0	0	0	0
5	0	0	0	3.9	0	0	0	0	0	0	0	0	0
6	0	0	0	0.0	0	0	0	0	0	0	0	0	0

	dose29	dose30	dose31	dose32
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0

The maximum number of exposures in this cohort is 32 ( $\text{max.exp} = 32$ )

## The package `fit.linERR`

- Profile likelihood based confidence intervals can be computed by means of function `ERRci`

### `ERRci`

The function `ERRci` needs the following arguments:

- `object`: An object of class `fit.linERR`
- `prob`: Level of confidence. Its default value is 0.95

## The package `fit.linERR`

- Profile likelihood based confidence intervals can be computed by means of function `ERRci`

### `ERRci`

The function `ERRci` needs the following arguments:

- `object`: An object of class `fit.linERR`
- `prob`: Level of confidence. Its default value is 0.95

## The package `fit.linERR`

- Profile likelihood based confidence intervals can be computed by means of function `ERRci`

### `ERRci`

The function `ERRci` needs the following arguments:

- `object`: An object of class `fit.linERR`
- `prob`: Level of confidence. Its default value is 0.95

## The package `fit.linERR`

- The standard procedure for computing a confidence interval for a parameter  $\beta$  (Wald-type CI) is based on

$$\hat{\beta} \pm z_{1-\frac{\alpha}{2}} SE(\hat{\beta}) \quad (1)$$

- It may work poorly if the distribution of the parameter estimator is markedly skewed or if the standard error is a poor estimate of the standard deviation of the estimator
- Profile likelihood confidence intervals don't assume normality of the estimator and perform better for small sample sizes or skewed estimates than Wald-type CI's

## The package `fit.linERR`

- The standard procedure for computing a confidence interval for a parameter  $\beta$  (Wald-type CI) is based on

$$\hat{\beta} \pm z_{1-\frac{\alpha}{2}} SE(\hat{\beta}) \quad (1)$$

- It may work poorly if the distribution of the parameter estimator is markedly skewed or if the standard error is a poor estimate of the standard deviation of the estimator
- Profile likelihood confidence intervals don't assume normality of the estimator and perform better for small sample sizes or skewed estimates than Wald-type CI's

## The package `fit.linERR`

- The standard procedure for computing a confidence interval for a parameter  $\beta$  (Wald-type CI) is based on

$$\hat{\beta} \pm z_{1-\frac{\alpha}{2}} SE(\hat{\beta}) \quad (1)$$

- It may work poorly if the distribution of the parameter estimator is markedly skewed or if the standard error is a poor estimate of the standard deviation of the estimator
- Profile likelihood confidence intervals don't assume normality of the estimator and perform better for small sample sizes or skewed estimates than Wald-type CI's



## The package `fit.linERR`

- The profile log-likelihood function can be plotted with the usual method `plot` applied to an object returned by `fit.linERR`, with an option to highlight the profile likelihood confidence intervals

### `plot`

The function `plot` needs the following arguments:

- `object`: An object of class `fit.linERR`
- `lower`: Lower value of the parameter
- `upper`: Upper value of the parameter
- `ci`: Highlight the profile likelihood confidence interval. Its default value is `NULL`

## The package `fit.linERR`

- The profile log-likelihood function can be plotted with the usual method `plot` applied to an object returned by `fit.linERR`, with an option to highlight the profile likelihood confidence intervals

### `plot`

The function `plot` needs the following arguments:

- `object`: An object of class `fit.linERR`
- `lower`: Lower value of the parameter
- `upper`: Upper value of the parameter
- `ci`: Highlight the profile likelihood confidence interval. Its default value is `NULL`

## The package `fit.linERR`

- The profile log-likelihood function can be plotted with the usual method `plot` applied to an object returned by `fit.linERR`, with an option to highlight the profile likelihood confidence intervals

### `plot`

The function `plot` needs the following arguments:

- `object`: An object of class `fit.linERR`
- `lower`: Lower value of the parameter
- `upper`: Upper value of the parameter
- `ci`: Highlight the profile likelihood confidence interval. Its default value is `NULL`

## The package `fit.linERR`

- The profile log-likelihood function can be plotted with the usual method `plot` applied to an object returned by `fit.linERR`, with an option to highlight the profile likelihood confidence intervals

### `plot`

The function `plot` needs the following arguments:

- `object`: An object of class `fit.linERR`
- `lower`: Lower value of the parameter
- `upper`: Upper value of the parameter
- `ci`: Highlight the profile likelihood confidence interval. Its default value is `NULL`

## The package `fit.linERR`

- The profile log-likelihood function can be plotted with the usual method `plot` applied to an object returned by `fit.linERR`, with an option to highlight the profile likelihood confidence intervals

### `plot`

The function `plot` needs the following arguments:

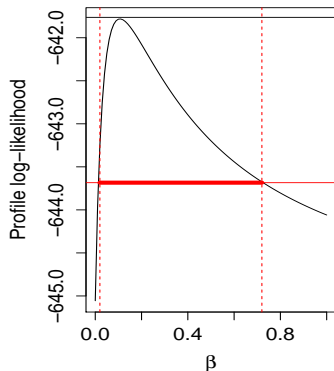
- `object`: An object of class `fit.linERR`
- `lower`: Lower value of the parameter
- `upper`: Upper value of the parameter
- `ci`: Highlight the profile likelihood confidence interval. Its default value is `NULL`

## The package `fit.linERR`

The command

```
> plot(fit.ex2, 0, 1, 0.95)
```

produces the plot



## Simulated cohorts without covariates

Three cohorts with a different number of cases and 10000 subjects were generated, and a linear excess relative risk model was fitted using the code

```
> fit.ex1 <- fit.linERR(ex1, beta=NULL, id=ex1[, 1], max.exp=32,
+                       status=ex1$leu, start=ex1$entryage,
+                       end=ex1$exitage, ages=ex1[, 7:38],
+                       doses=ex1[, 39:70])
```

	Cases	$\beta$	$\hat{\beta}$ (Epicure)	$\hat{\beta}$ (R)	Computing time (Epicure)	Computing time (R)
Cohort 1	13	0.04	0.07213 (0.1714)	0.07140 (0.1374)	11.64	2.26
Cohort 2	19	0.1	0.3745 (0.8677)	0.3795 (0.8165)	27.34	3.97
Cohort 3	14	0	-0.0242 (NA)	-0.0242	10.47	0.4

Table: Results from Epicure and R.

## Simulated cohort with covariates and strata

A cohort consisting of 150000 subjects was generated, and a linear excess relative risk model including three covariates and different strata was fitted using the code

```
> fit.ex2 <- fit.linERR(ex2, beta=NULL, id=ex2[, 2], max.exp=32,
+                       status=ex2$leu, start=ex2$entryage,
+                       end=ex2$exitage, ages=ex2[, 9:40],
+                       doses=ex2[, 41:72],
+                       covariates1=data.frame(ex2$ds, ex2$ds2, ex2$ds3),
+                       strata=ex2[,1])
```

	Cases	$\beta$	$\hat{\beta}$ (Epicure)	$\hat{\beta}$ (R)	Computing time (Epicure)	Computing time (R)
Cohort 1	84	0.1	0.1084 (0.09263)	0.1077 (0.0866)	176.8	123.74

Table: Results from Epicure and R.



## Simulated cohort with covariates and strata

The ML estimates can be obtained by

```
> summary(fit.ex2)
...
```

Parameter Summary Table:

	Estimate	Std. Error	Test Stat.	p-value
dose	0.10764506	0.086609701	1.2428753	2.139138e-01
ex2\$ds	4.35150398	0.281627097	15.4512972	7.394023e-54
ex2\$ds2	0.04891001	0.245572672	0.1991671	8.421320e-01
ex2\$ds3	-0.00212656	0.008645206	-0.2459814	8.056966e-01

AIC: 1291.526

Deviance: 1283.526

## Simulated cohort with covariates and strata

The 95% profile likelihood-based confidence interval can be computed with the command

```
> ERRci(fit.ex2)
  lower 2.5% upper 97.5%
0.01352891  0.72524683
```

## Simulation study



A simulation study to evaluate the effects of various issues on estimated risks of cancer due to medical exposure to ionising radiation using synthetic cohorts will be carried on within the framework of the project Epi-CT using the package `fit.linERR`

- Reverse causation
- Confounding by indication
- Missing data
- Missing doses





CREAL<sup>9</sup>

Centre for Research  
in Environmental  
Epidemiology



Parc de Recerca Biomèdica de Barcelona  
Doctor Aiguader, 88  
08003 Barcelona (Spain)  
Tel. (+34) 93 214 70 00  
Fax (+34) 93 214 73 02

info@creal.cat  
www.creal.cat

Universitat Autònoma de Barcelona  
www.uab.cat

**ISGlobal**  
Barcelona  
Institute for  
Global Health

And its research centres



**CRESIB<sup>9</sup>**  
BARCELONA

Institut Ciències Comercials de Barcelona



centre for research  
in environmental  
epidemiology