# R program for flexible Cox models including time-dependent (TD) and/or non-linear (NL) effects: `CoxFlex`

Marie-Eve Beauchamp[1] and Yishu Wang[2]

Program by: Willy Wynant[3], Yishu Wang[2], Marie-Eve Beauchamp[1]

[1] Centre for Outcomes Research and Evaluation, Research Institute of the McGill University Health Centre, Montreal, Canada

[2] Scotia Bank, Toronto, Canada

[3] Syneos Health, Montreal, Canada

March 15, 2023

# R script

- Function **`CoxFlex`** allows to estimate a Cox model ***with time-dependent (TD) and/or non-linear (NL) effects*** for ***one or several variables***
  - Can include variables without TD and NL effects

- `CoxFlex` can handle:

  *a)* *Time-invariant data* (one observation per subject)

| id | time | dose | event |
|---|---|---|---|
| 1 | 56 | 3.0 | 1 |
| 2 | 365 | 0.5 | 0 |
| 3 | 283 | 0 | 0 |

  *b)* *Time-dependent* (or time-varying) data (several observations per subject)

| id | start | stop | dose | event |
|---|---|---|---|---|
| 1 | 0 | 14 | 1.0 | 0 |
| 1 | 14 | 28 | 2.0 | 0 |
| 1 | 28 | 56 | 3.0 | 1 |
| 2 | 0 | 180 | 1.0 | 0 |
| 2 | 180 | 365 | 0.5 | 0 |

# Data preparation

1. The data must be a **data frame**

2. The **first column of data *must* be a numeric ID variable** identifying the subjects (with the name of your choice)

3. **No missing data** are allowed (otherwise the function will crash)

4. All string characters or factors *must* be recorded as numeric values
   - E.g., gender: (0, 1) instead of (“male”, “female”)

5. All binary variables must be coded (0, 1)

# Data preparation

5. Categorical variables, with more than 2 categories, *must* be recorded as dummy variables (not as factor)
   – E.g., if 4 age groups (<18, 18-39, 40-64, ≥65) with reference <18, then create 3 dummy (binary) variables to include in the model

| age.gr |
|---|
| <18 |
| 18-39 |
| 40-64 |
| ≥65 |

| age18.39 | age40.64 | age65 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

6. Negative values of continuous covariates are *not* a problem, as opposed to when using fractional polynomials

7. Include in dataset passed to `CoxFlex` *only* the variables used in the model. This will greatly improve the efficiency (speed) of the program.

# Data preparation

8. <u>For time-varying data:</u>

   – Each line can be for time **intervals with length of 1** (e.g. 1 day) ***or longer***

   – The 'start' of a line must be the same as the 'stop' of the previous line (for the same subject), i.e. **no gap and no overlap in time intervals**

   – No intervals with 'start' = 'stop'

| id | start | stop | dose | event |
|---|---|---|---|---|
| 1 | 0 | 14 | 1.0 | 0 |
| 1 | 14 | 28 | 2.0 | 0 |
| 1 | 28 | 56 | 3.0 | 1 |
| 2 | 0 | 180 | 1.0 | 0 |
| 2 | 180 | 365 | 0.5 | 0 |

   – The 1$^{st}$ start value of each subject *must* be 0 (delayed entry not allowed)

# Data preparation

9. For time-invariant data (1 line per subject):
   – Event times must be > 0

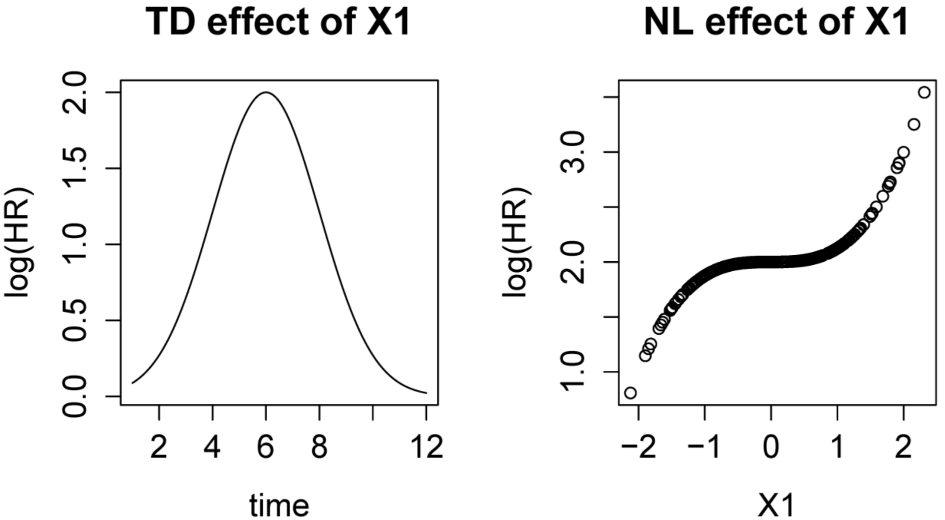
# Example of a dataset (`dat`) with 3 time-varying covariates (available with the R script)

| Id | Event | Fup | Start | Stop | x1 | x2 | x3 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 2 | 0 | 1 | -0.23549 | 1.541651 | 0.87255 |
| 1 | 1 | 2 | 1 | 2 | -0.86123 | -1.27724 | 0.87255 |
| 2 | 0 | 2 | 0 | 1 | 0.737676 | -0.17344 | 1.959211 |
| 2 | 1 | 2 | 1 | 2 | 1.345106 | -0.60027 | 1.959211 |
| 3 | 0 | 8 | 0 | 1 | 0.863421 | 1.182985 | 3.862302 |
| 3 | 0 | 8 | 1 | 2 | 0.914108 | 0.784081 | 3.862302 |
| 3 | 0 | 8 | 2 | 3 | 0.933757 | -0.38651 | 3.862302 |
| 3 | 0 | 8 | 3 | 4 | 1.19725 | -0.77683 | 3.862302 |
| 3 | 0 | 8 | 4 | 5 | 1.139094 | 1.371349 | 3.862302 |
| 3 | 0 | 8 | 5 | 6 | 0.518455 | 0.056261 | 3.862302 |
| 3 | 0 | 8 | 6 | 7 | 0.575675 | -0.71817 | 3.862302 |
| 3 | 1 | 8 | 7 | 8 | 0.632166 | 0.67002 | 3.862302 |

# Example

- 300 patients followed for up to 12 months (in `dat`)
- $X_1(t)$, $X_2(t)$, $X_3(t)$ are continuous time-varying variables
- **True model** (data generated from it):
  - $X_1$: TD and NL effects
  - $X_2$ and $X_3$: constant-over-time and linear effects

$$\lambda(t \mid X_1(t), X_2(t), X_3(t)) = \lambda_0 \exp\{ \beta_1(t) g_1(X_1(t)) + \beta_2 X_2(t) + \beta_3 X_3(t) \}$$



# Download CoxFlex R script & data

- https://github.com/mebeauchamp/CoxFlex
  - Content:
    - **CoxFlex - 20220330 - to share.R**: R script including CoxFlex and related functions
    - **dat.Rdata**: Dataset used for the example in the tutorial

# Code

```
# Source the program in current R session (not a package yet):
source("C:/.../CoxFlex - 20220330 - to share.R")
   # No need to look at the code in this file

# Load data
load("C:/.../dat.RData")

head(dat)
  # Id Event Fup Start Stop         x1         x2        x3
  #  1     0   2     0    1 -0.2354852  1.5416514 0.8725503
  #  1     1   2     1    2 -0.8612346 -1.2772441 0.8725503
  #  2     0   2     0    1  0.7376760 -0.1734419 1.9592107
  #  2     1   2     1    2  1.3451063 -0.6002743 1.9592107
  #  3     0   8     0    1  0.8634209  1.1829845 3.8623023
  #  3     0   8     1    2  0.9141075  0.7840808 3.8623023

# Select only variables relevant for model estimation
dat.red <- dat[, c('Id','Event','Start','Stop','x1','x2','x3')]
```

```r
# Check if data are in a data frame
is.data.frame(dat.red)
    # [1] TRUE

# Check if the ID variable is numeric (must be 1st column)
is.numeric(dat.red[, 1])
    # [1] TRUE

# Display structure of data (all variables must be numeric, 1st one is ID)
str(dat.red)
 # 'data.frame':   2307 obs. of  7 variables:
 #  $ Id   : num  1 1 2 2 3 3 3 3 3 3 ...
 #  $ Event: num  0 1 0 1 0 0 0 0 0 0 ...
 #  $ Start: num  0 1 0 1 0 1 2 3 4 5 ...
 #  $ Stop : num  1 2 1 2 1 2 3 4 5 6 ...
 #  $ x1   : num  -0.235 -0.861 0.738 1.345 0.863 ...
 #  $ x2   : num  1.542 -1.277 -0.173 -0.6 1.183 ...
 #  $ x3   : num  0.873 0.873 1.959 1.959 3.862 ...

# Check no missing values in any variables used for the model
sum(is.na(dat.red))
    # [1] 0

# Distribution of 1st Start value across subjects (check no delayed entry)
table(by(dat.red$Start, dat.red$Id, min))
    #   0
    # 300
```

# Estimation of a *predefined* model with `CoxFlex`

```
m1 <- CoxFlex(data=dat.red, Type=c("Start","Stop","Event"),
              variables=c("x1","x2","x3"),
              TD=c(1,0,0), NL=c(1,0,0),
              m=1, p=2, knots=-999)
```

**ARGUMENTS:**

- `data`: Your dataset (data frame). *1st column must be an ID variable of individuals.*
- `Type`: Variables in data indicating the start and stop of time intervals, and the event (1=event, 0=censored).
  **If time-varying data:** `Type=c("Start","Stop","Event")`.
  **If time-invariant data:** `Type=c("Time","Event")`.
  Start, Stop, and Time do *not* have to be integers.
- `variables`: All independent variables in the model (exposure and covariates)
- `TD`: Indicate for each independent variable if the TD effect is modeled (0/1)
- `NL`: Indicate for each independent variable if the NL effect is modeled (0/1).
  *Can be 1 only for continuous variables.*

# Estimation of a *predefined* model with `CoxFlex`

```
m1 <- CoxFlex(data=dat.red, Type=c("Start","Stop","Event"),
              variables=c("x1","x2","x3"),
              TD=c(1,0,0), NL=c(1,0,0),
              m=1, p=2, knots=-999)
```

**ARGUMENTS:**

- `m`: Number of interior knots (the same for all TD and NL effects). By default **`m=1`**.
- `p`: Order of splines (the same for all TD and NL effects). By default **`p=2`**.
  - p=0: step functions
  - p=1: linear splines
  - p=2: quadratic splines
  - p=3: cubic splines
- `knots`: Position of interior knots. Default **`knots=-999`**, which indicates that the knots are automatically allocated.

  To specify the position of interior knots, specify a matrix with (`length(variables)+1`) rows by `m` columns. There is one row per variable (add NA if no NL effect for a variable) and one for time. E.g., for this model it could be:

  `knots = matrix(c(-1,NA,NA,4), nrow=4, ncol=1).`

# Output of the model (more on next slides)

```
# Type the name of the object of results to see the output
m1

$Partial_Log_Likelihood
[1] -1013.063

$Number_of_parameters
[1] 9

$Number_events
[1] 202

$Number_knots
[1] 1

$Degree_of_splines
[1] 2
```

To calculate AIC use:

AIC =

```
-2 * m1$Partial_Log_Likelihood
+ 2 * m1$Number_of_parameters
```

```
$knots_covariates
        [,1]      [,2]      [,3]        [,4]
x1 -3.229684 -3.229684 -3.229684 -0.06910736
x2        NA        NA        NA          NA
x3        NA        NA        NA          NA
       [,5]     [,6]     [,7]
x1 3.687497 4.687497 5.687497
x2       NA       NA       NA
x3       NA       NA       NA

$knots_time
[1]  0  0  0  5 12 13 14
```

Position of interior and exterior knots, for each variable with a NL effect

Position of interior and exterior knots for time

```
$coefficients
       x1         x2         x3
       NA 0.3267542 0.1293491

$Standard_Error
[1]    NA 0.07258465 0.01504884

$coefficients_splines_NL
             x1 x2 x3
[1,] 0.000000 NA NA
[2,] 2.568096 NA NA
[3,] 2.445147 NA NA
[4,] 6.327139 NA NA

$coefficients_splines_TD
               x1 x2 x3
[1,]  0.5779502 NA NA
[2,]  1.3659911 NA NA
[3,]  1.5177027 NA NA
[4,] -0.4379260 NA NA
```

**Coefficients (log hazard) and SE for variables *<u>without</u>* TD nor NL effects**

Coefficients of splines for NL and TD effects requested:

3 splines for a NL effect (m+p):
First NL spline coefficient always set 0 for technical reasons.

4 splines for a TD effect (m+p+1).

```
$variables
[1] "x1" "x2" "x3"
```

```
$coef
[1] NA       NA       0.327    0.129

$var
[1] NA       NA       0.005329 0.000225

$pvalue
[1] 0.264    0.398    0.000    0.000
```

```
$variables
[1] "x1" "x2" "x3"
```

For <u>each variable</u> above, the values shown below are, respectively, for:

1) ***NL* effect (when applicable), and/or**
2) ***TD* effect (when applicable), or**
3) **"Standard" effect when no NL nor TD effects were requested.**

Then, move to the next variable.

For this model, the request was:

`variables=c("x1","x2","x3"), TD=c(1,0,0), NL=c(1,0,0)`

Therefore, the values reported below are for:

| | NL x1 | TD x1 | x2 | x3 |
|---|---|---|---|---|
| $coef [1] | NA | NA | 0.327 | 0.129 |
| $var [1] | NA | NA | 0.005329 | 0.000225 |
| $pvalue [1] | 0.264 | 0.398 | 0.000 | 0.000 |

```
$variables
[1] "x1" "x2" "x3"
```

**Likelihood ratio tests (LRTs):**

| NL x1: | TD x1: | x2: | x3: |
|---|---|---|---|
| Linear x1 vs. NL x1 | PH x1 vs. TD x1 | $\beta_2=0$ vs. $\beta_2\neq0$ | $\beta_3=0$ vs. $\beta_3\neq 0$ |

```
$pvalue
[1] 0.264    0.398   0.000     0.000
```

NL and TD effects of x1 are non-significant. Does it mean x1 is not important?

# Standard Cox PH model

```
library(survival)
m.cox <- coxph(Surv(Start, Stop, Event) ~ x1 + x2 + x3, data=dat.red)
m.cox
   #Call:
   #coxph(formula = Surv(Start, Stop, Event) ~ x1 + x2 + x3, data = dat.red)

   #    coef exp(coef) se(coef)    z       p
   #x1 0.401      1.49   0.0728 5.51 3.6e-08
   #x2 0.316      1.37   0.0721 4.38 1.2e-05
   #x3 0.136      1.15   0.0151 8.99 0.0e+00

   #Likelihood ratio test=89.1  on 3 df, p=0  n= 2307, number of events= 202

# Significant effect for x1, even though the NL and TD effects were not
# significant in the flexible model.
# Don't discard a variable because NL and/or TD effects are non-significant!

AIC(m.cox)
   # [1] 2037.618
BIC(m.cox)
   # [1] 2047.543
```

# AIC/BIC for a model estimated with CoxFlex

```
# AIC
-2 * m1$Partial_Log_Likelihood + 2 * m1$Number_of_parameters
   # [1] 2044.127

# BIC
-2 * m1$Partial_Log_Likelihood +
  log(m1$Number_events) * m1$Number_of_parameters
   # [1] 2073.901

# Both AIC and BIC are higher (worse) than for standard Cox model,
# confirming the extra parameters to model NL and TD effects of x1 did
# not improve the fit to the data
```

# Order of p-values in `$pvalue` for another example of model

- If the model requested in the CoxFlex function was:

**`variables=c("x1","x2","x3"), TD=c(0,1,0), NL=c(0,1,1)`**

For <u>each variable</u>, the p-values shown are, respectively, for:

***1)*** ***NL* effect (when applicable), and/or**
***2)*** ***TD* effect (when applicable), or**
**3) "Standard" effect when no NL nor TD effects were requested.**

Then, move to the next variable.

- Then, the p-values in vector $pvalue would be for:
  - Significance of "standard" effect for x1 ($\beta_1 \neq 0$)
  - <u>*NL*</u> effect of x2
  - <u>*TD*</u> effect of x2
  - NL effect of x3

# Plot the NL/TD effects: plot.FlexSurv

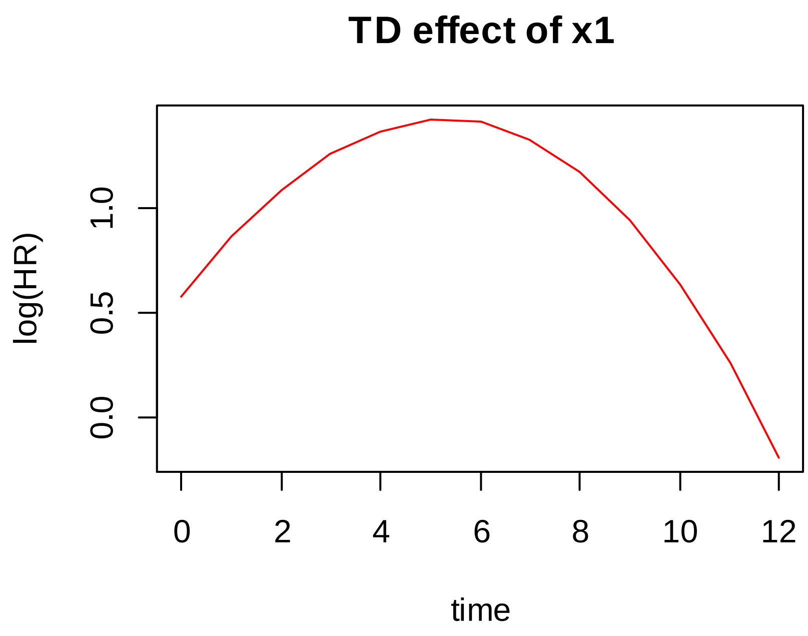
```
# To plot two graphs on top of each other
par(mfrow=c(2,1))
```

Only 1 variable at the time

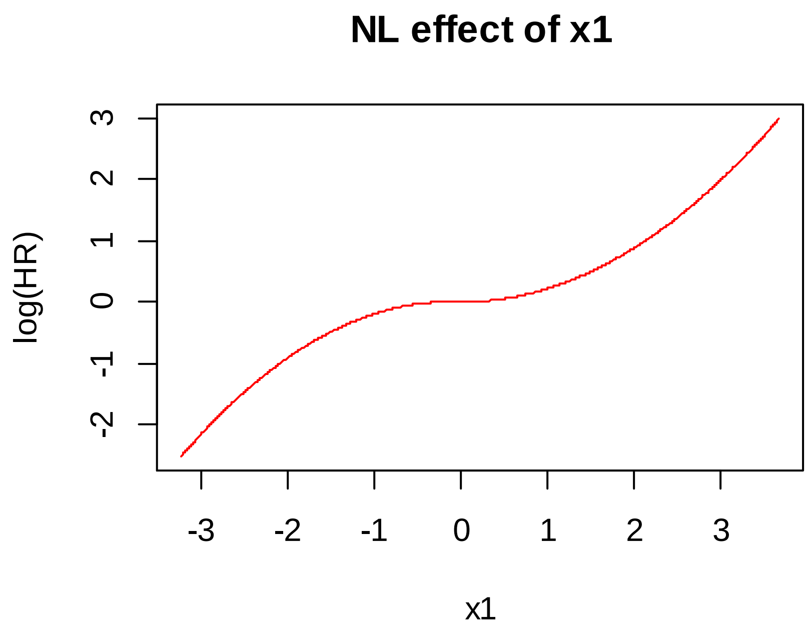
```
# Plot for TD effect of x1
plot.FlexSurv(model.FlexSurv = m1, variable="x1", TD=1, NL=0,
   col="red", xlab="time", ylab="log(HR)",
   main="TD effect of x1", type="l")
```

NL effect plotted with respect to *this reference value* of the variable (default 0)

```
# Plot for NL effect of x1
plot.FlexSurv(model.FlexSurv = m1, variable="x1", TD=0, NL=1, ref.value.NL=0,
   col="red", xlab="x1", ylab="log(HR)",
   main="NL effect of x1", type="l")
```



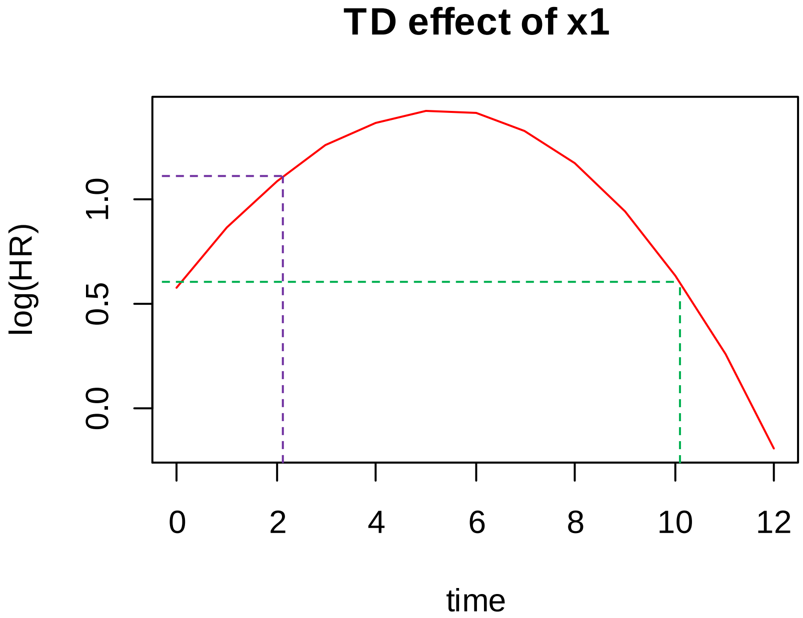

Shows how the *strength* of the effect of x1 varies over time.




Shows the log(HR) comparing each value of x1 (numerator) to the reference value x1=0 (denominator)

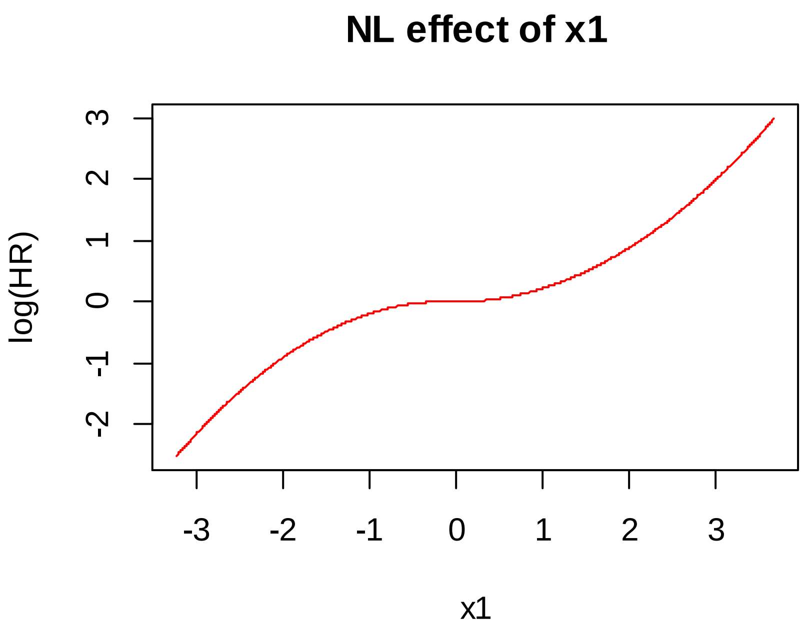
However, in the *current model estimated*, TD and NL effects for x1 are *multiplied* by each other: $\beta_1(t) \cdot g_1(X_1(t))$

Consequently, *shapes* of NL and TD effects are good on these independent graphs, but *not* log(HR) on y axes




E.g., at $t$=2 the NL effect of x1 has to be multiplied by 1.1

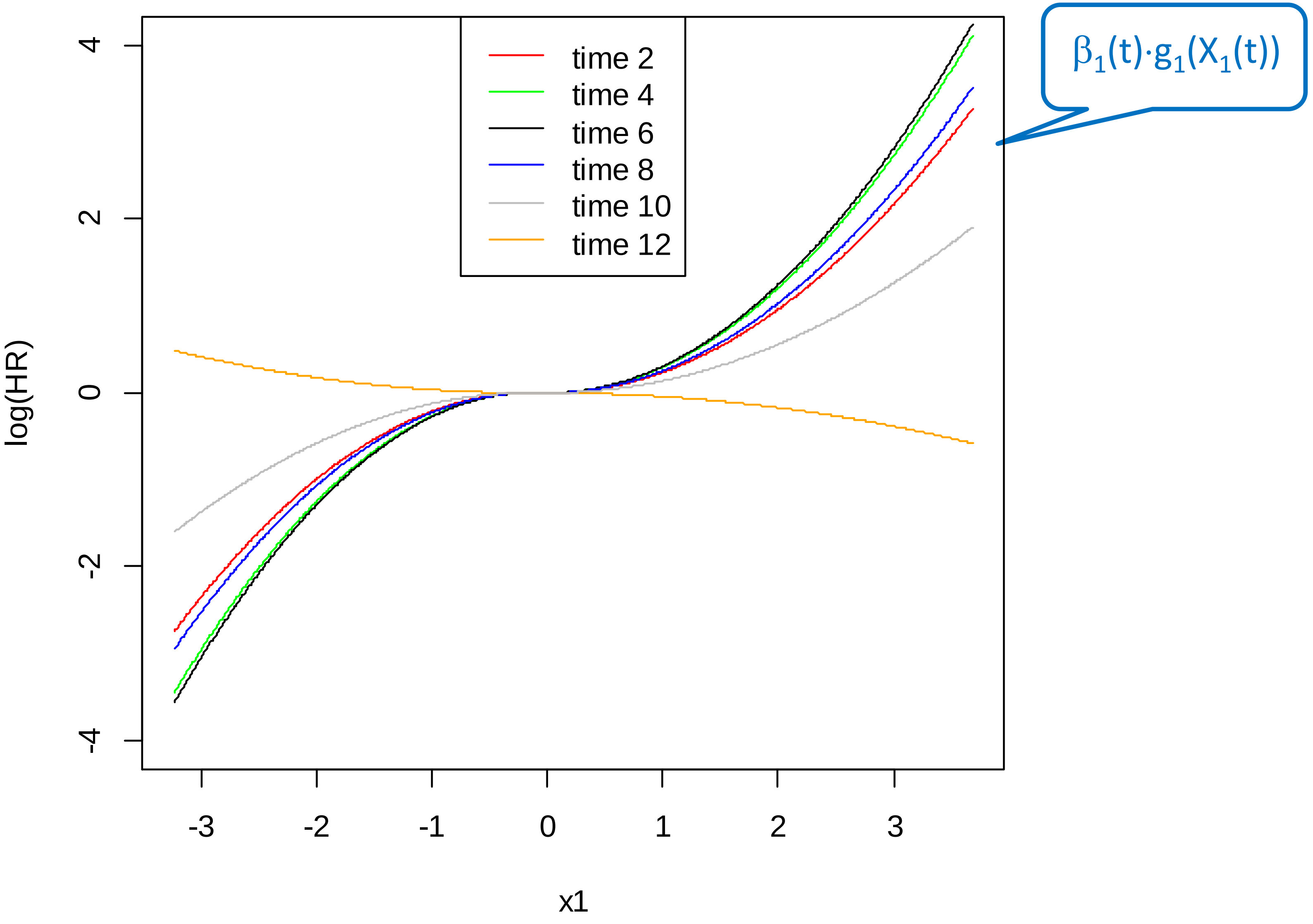
But at $t$=10 the NL effect has to be multiplied by 0.6



# NL effect at fixed time points (when a TD effect is also modeled)

```
par(mfrow=c(1,1))

plot.FlexSurv(m1, variable="x1", TD=1, NL=1, TimePoint=2, ref.value.NL=0,
   ylim=c(-4,4), xlab="x1", ylab="log(HR)", type="l", col="red",
   main="Total effect: NL effect of x1 at fixed time points")

lines.FlexSurv(m1, variable="x1", TD=1, NL=1, TimePoint=4, ref.value.NL=0,
   col="green")

lines.FlexSurv(m1, variable="x1", TD=1, NL=1, TimePoint=6, ref.value.NL=0,
   col="black")
lines.FlexSurv(m1, variable="x1", TD=1, NL=1, TimePoint=8, ref.value.NL=0,
   col="blue")
lines.FlexSurv(m1, variable="x1", TD=1, NL=1, TimePoint=10, ref.value.NL=0,
   col="gray")
lines.FlexSurv(m1, variable="x1", TD=1, NL=1, TimePoint=12, ref.value.NL=0,
   col="orange")

legend("top", c("time 2","time 4","time 6","time 8","time 10","time 12"),
   lty=c(1,1,1,1,1,1), col=c("red","green","black","blue","gray","orange"))
```

# Total effect: NL effect of x1 at fixed time points



# Backward selection of NL/TD effects

```
m2 <- backward_selection2(data=dat.red,
        Type=c("Start","Stop","Event"),
        variables=c("x1","x2","x3"),
        continuous=c(1,1,1),
        TD=c(0,0,0), NL=c(0,0,0),
        m=1, p=2, alpha_back=0.05, knots=-999)
```

**ARGUMENTS:**

- `continuous`: Indicate whether each variable is continuous (1=yes, 0=no)
- `TD=1 / NL=1`: Force TD/NL effect of corresponding variable
- `TD=0 / NL=0`: Do not force the effect (TD/NL effects are evaluated, and a variable may be excluded from final model)
- `TD=-1 / NL=-1`: Exclude TD/NL effect of corresponding variable (i.e. force the PH/LL effect)
- `alpha_back`: Alpha value used to select effects

```
# Command to see model output for the final model
m2
$final_model$Partial_Log_Likelihood
[1] -999.011

$final_model$Number_of_parameters
[1] 5

$final_model$Number_events
[1] 202

$final_model$Number_knots
[1] 1

$final_model$Degree_of_splines
[1] 2

$final_model$knots_covariates
         [,1]       [,2]       [,3]     [,4]     [,5]     [,6]     [,7]
x1         NA         NA         NA       NA       NA       NA       NA
x2         NA         NA         NA       NA       NA       NA       NA
x3 0.04792073 0.04792073 0.04792073 0.846845 35.50922 36.50922 37.50922

$final_model$knots_time
[1]  0  0  0  5 12 13 14
```

Here, better likelihood, with fewer parameters, than the predefined model (-1013.063, with 9 parameters)

```
$final_model$coefficients
       x1         x2         x3
0.3743083 0.3481198         NA

$final_model$Standard_Error
[1] 0.07211582 0.07257598         NA

$final_model$coefficients_splines_NL
     x1 x2         x3
[1,] NA NA 0.0000000
[2,] NA NA 0.2563118
[3,] NA NA 7.4415084
[4,] NA NA 3.0269701

$final_model$coefficients_splines_TD
     x1 x2 x3
[1,] NA NA NA
[2,] NA NA NA
[3,] NA NA NA
[4,] NA NA NA
```

NL effect selected for x3,

but no TD effects

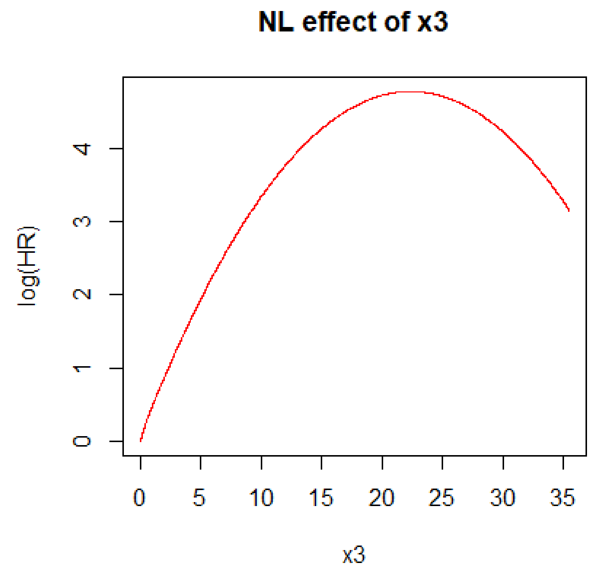
```
$final_model$variables
[1] "x1" "x2" "x3"

$final_model$coef
[1] 0.374 0.348    NA

$final_model$var
[1] 0.005184 0.005329       NA

$final_model$pvalue
[1] 0 0 0
```

Showing respectively, p-value for significance of:

- $\beta_1 = 0$ vs. $\beta_1 \neq 0$
- $\beta_2 = 0$ vs. $\beta_2 \neq 0$
- Linear x3 vs. NL x3

# Model selected by backward selection

```
# AIC
-2 * m2$final_model$Partial_Log_Likelihood + 2 * m2$final_model$Number_of_parameters
   # [1] 2008.022
   # Better than standard Cox (2037.6)

plot.FlexSurv(model.FlexSurv = m2$final_model, variable="x3", TD=0, NL=1,
         ref.value.NL = min(dat.red$x3),
         col="red", xlab="x3", ylab="log(HR)",
         main="NL effect of x3", type="l")
```



# Estimation problem: **NL effect** of a variable with ***non-negative values & frequent 0 values***

- Variable $X$ with all values $\geq 0$ but with 0 for a majority of observations (i.e. median($X$) = 0). E.g.,
  - Number of cigarettes per day, when > 50% of non-smokers
  - Drug dose, when subjects are often unexposed
- Estimation problem for NL effect: the interior knot is placed at median($X$) = 0, which is also min($X$)
  - Spline estimation crashes because interior knot = one of the exterior knots
- The problem would also occur if median($X$) = max($X$)
- But *no problem* if only a TD effect is requested for $X$

# Solution

1. Create a new binary variable ***Z:***

   ***Z*=1 if *X*=0**

   ***Z*=0 if *X>0***

1. Create a new variable ***X.centered***:
   – Center the *non-zero* values of original *X* at 0, i.e. subtract the mean of *non-zero X* values (say **M**) to each *non-zero* value of *X*
   – Keep *X.centered*=0 when original *X*=0
   ➢ Therefore, mean(*X.centered*) = median(*X.centered*) = 0, but min(*X.centered*) < 0

2. Estimate the flexible model with:

   NL effect of ***X.centered*** (excluding original *X*) + ***Z*** + all other covariates

   ➢ **Now, the NL effect describes the NL effect of non-zero values of *X*, while HR of *Z* estimates the HR for *X*=M vs. *X*=0**

NOTES: You can include TD effects for *X.centered* and *Z* in the model

# Explanation of effects

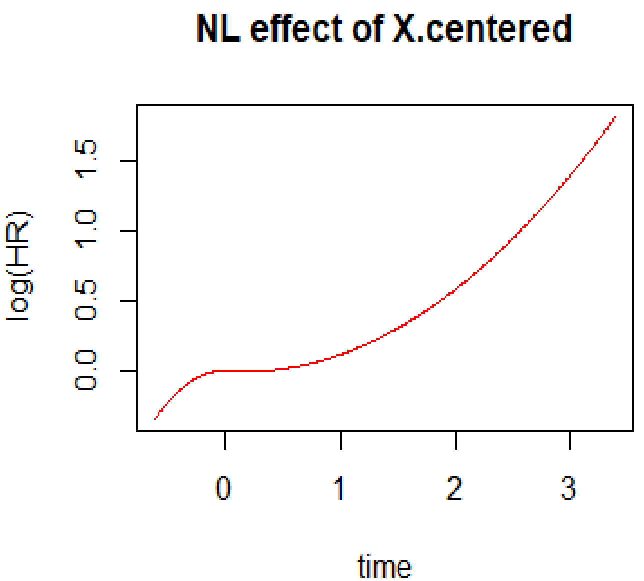
- **<u>HR of Z</u>** estimates the HR for *X*=M vs. *X*=0:
  - HR for Z = $\lambda(t|Z=1, X.centered, \text{covariates}) \,/\, \lambda(t|Z=0, X.centered, \text{covariates})$
    - i.e. hazard ratio for *Z*=1 vs. *Z*=0 *while keeping constant other variables in the model, including X.centered*
    - Occurs only for *X.centered* = M that *Z* could equal 1 or 0
  - Thus, hazard ratio for Z=1 vs. Z=0 *when X.centered = M*

    $\Rightarrow$ Hazard ratio for *X*=M vs. *X*=0

- **<u>NL effect of *X.centered*</u>** describes the effect of non-zero values of *X:*
  - It's the NL effect of *X.centered, while keeping constant other variables in the model, including Z*
    - Occurs only for *Z*=0 that *X.centered* could change value, i.e. for non-zero values of *X*

# Code (on mock data)

```
summary(data$X)
  # Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  # 0.0000  0.0000  0.0000  0.6597  1.0000  5.0000

M <- mean(data$X[data$X != 0])
data$X.centered <- ifelse(data$X == 0, 0, data$X - M)
summary(data$X.centered)
  # Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  # -0.614  -0.614   0.000   0.000   0.000   3.386

dat.red$Z <- ifelse(dat.red$X > 0, 1, 0)

m3 <- CoxFlex(data=data,
       Type=c("Start","Stop","Event"),
       variables = c("X.centered", "Z", "x1", "x2"),
       TD=c(1,1,0,0), NL=c(1,0,1,0),
       m=1, p=2, knots=-999)
```



# References

*References to cite:*


- Abrahamowicz M, MacKenzie TA. Joint estimation of time-dependent and non-linear effects of continuous covariates on survival. *Statistics in Medicine* 2007;26(2):392-408.
- Wynant W, Abrahamowicz M. Impact of the model-building strategy on inference about nonlinear and time-dependent covariate effects in survival analysis. *Statistics in Medicine 2014; 33: 3318–3337.*


*Examples of applications:*


- Gagnon B, Abrahamowicz M, Xiao Y, Beauchamp ME, MacDonald N, Kasymjanova G, Kreisman H, Small D. Flexible modeling improves assessment of prognostic value of C-reactive protein in advanced non-small cell lung cancer. *British Journal of Cancer* 2010;102(7):1113-1122.
- Le Teuff G, Abrahamowicz M, Wynant W, Binquet C, Moreau M, Quantin C. Flexible modeling of disease activity measures improved prognosis of disability progression in relapsing-remitting multiple sclerosis. *J Clin Epidemiol* 2015;68(3):307-16.
- Isidean SD, Wang Y, Mayrand M-H, Ratnam S, Coutlée F, Franco EL, Abrahamowiz M, for the CCCaST Study Group. Assessing the time-dependence of prognostic values of cytology and human papillomavirus testing in cervical cancer screening. *Int J Cancer* 2019;144(10):2408-2418.

# Help!

- ***For conceptual questions about your project***:

  Prof. Michal Abrahamowicz

- ***To obtain the program, and for help with it***:

  marie-eve.beauchamp@rimuhc.ca