trajeR – une nouvelle librairie R pour les modèles de mélanges pour données longitudinales.

Cédric NOEL[1], Jang SCHILTZ[2]

[1] IUT de Thionville-Yutz, Université de Lorraine, Espace Cormontaigne Impasse Alfred Kastler F-57970 Yutz, France - Université du Luxembourg
[2]Département de Finance, Université du Luxembourg, 6, rue Richard Coudenhove-Kalergi L-1359 Luxembourg, Luxembourg

UNIVERSITÉ DE LORRAINE | IUT THIONVILLE-YUTZ

UNIVERSITÉ DU LUXEMBOURG

Présentation du modèle

On considère des trajectoires de différents individus dont on suppose qu'elles se répartissent en plusieurs groupes homogènes. La méthode "group-based trajectory modeling" (GBTM) recherche ces groupes ainsi que la trajectoire moyenne de chaque groupe.

Exemple illustratif

Données issues de Jones et Nagin

`https://www.andrew.cmu.edu/user/bjones/cnorm.htm`

Exemple illustratif

Données issues de Jones et Nagin

`https://www.andrew.cmu.edu/user/bjones/cnorm.htm`

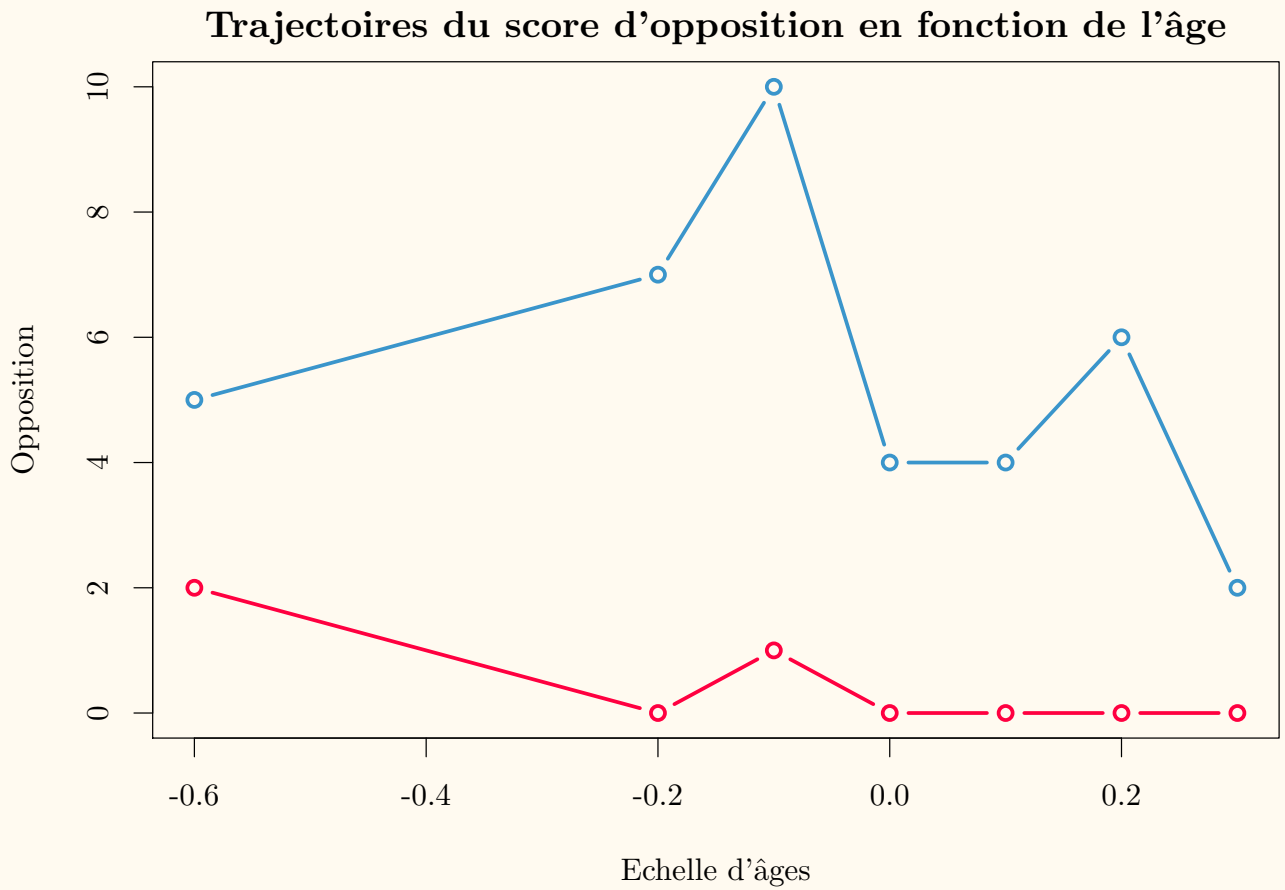
138 enfants issus de "l'étude longitudinale de Montréal" (Tremblay et al.).

Exemple illustratif

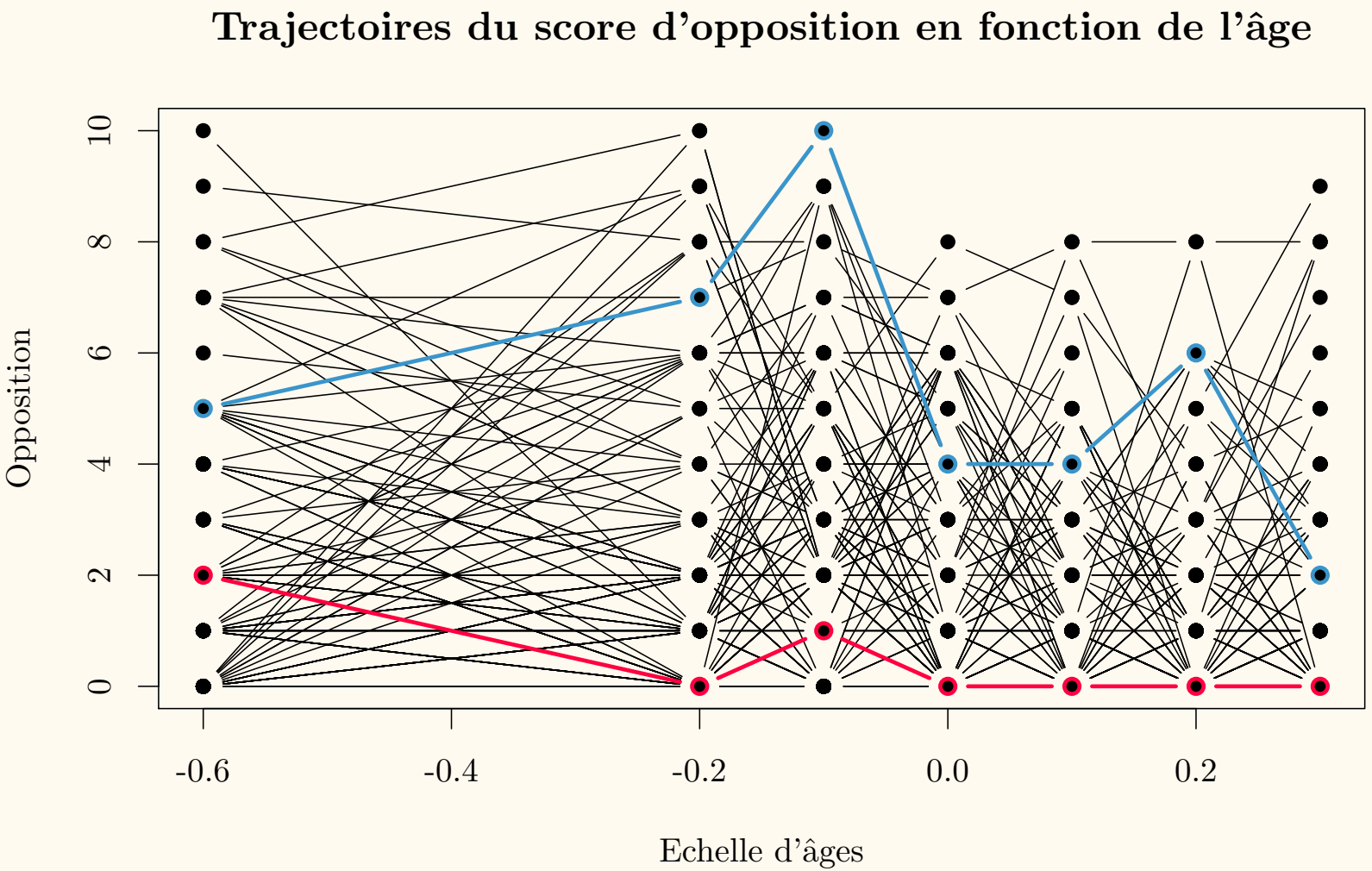
Données issues de Jones et Nagin

`https://www.andrew.cmu.edu/user/bjones/cnorm.htm`

138 enfants issus de "l'étude longitudinale de Montréal" (Tremblay et al.).



Exemple illustratif



Présentation du modèle

Soit $Y_i = \{y_{i_1}, y_{i_2}, ..., y_{i_T}\}$, T mesures de la variable Y, à différents temps $t_1, ..., t_T$ pour un sujet i.
Soit π_k la probabilité pour un individu donné d'appartenir au groupe k parmi K.

$$P(Y_i) = \sum_{k=1}^{K} \pi_k g_k(y_i; \Theta_k). \tag{1}$$

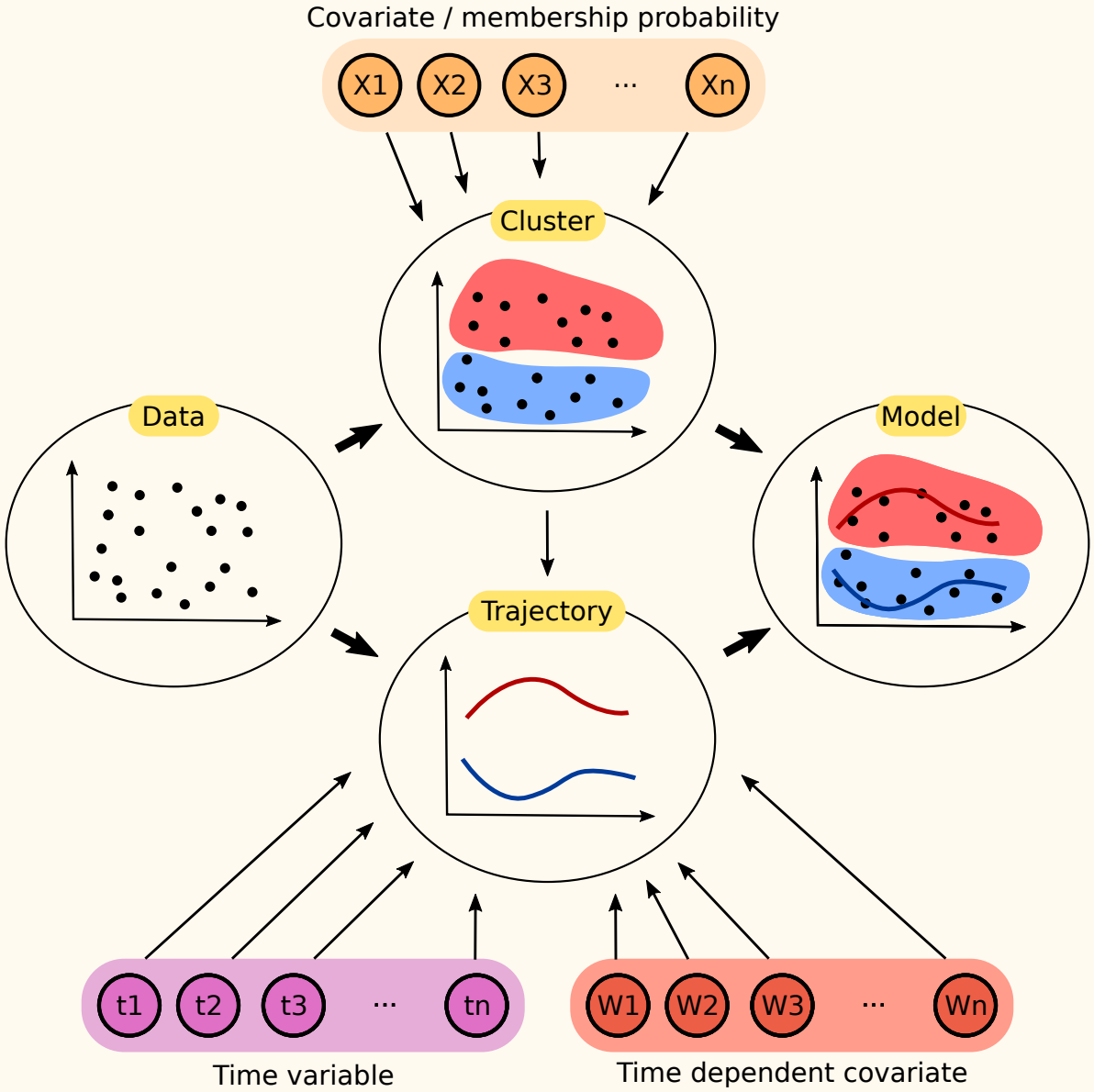
But : estimer les paramètres $\Omega = \{K, \pi_k, \Theta_k; k = 1, ..., K\}$.

Présentation du modèle

La densité g_k est influencée par une fonction du temps A_i qui est relié à la trajectoire par les paramètres β_k et parfois par une autre variable temporelle W_i avec les paramètres δ_k.

La probabilité d'appartenance à un groupe peut être influencée par une variable X_i avec des paramètres θ_k.

Présentation du modèle



Présentation du modèle

3 distributions pour modéliser Y :

- la loi logistique ;
- la loi ZIP (Zero Inflated Poisson) ;
- la loi normale censorée.

Loi logistique

Soit $\rho_{ikt} = P(Y_{it} = 1 | W_i = w_i, C_i = k)$ la probabilité que $y_{it} = 1$ étant donné l'appartenance à un groupe k.

$$\rho_{ikt} = \frac{e^{\beta_k A_{it} + \delta_k W_{it}}}{1 + e^{\beta_k A_{it} + \delta_k W_{it}}} \tag{2}$$

où $A_{it} = (1, a_{it}, a_{it}^2, \cdots, a_{it}^{n_\beta - 1})^t$, $W_t = (w_{i1}, \cdots, w_{in_\delta})^t$, $\beta_k = (\beta_{k1}, \cdots, \beta_{kn_\beta})$ and $\delta_k = (\delta_{k1}, \cdots, \delta_{kn_\delta})$

Loi Zero Inflated Poisson

La loi ZIP utilise deux processus différents : une loi binaire qui génère les zéros en excès, et une distribution de Poisson qui génère le comptage.

On a

$$P\left(Y_{it} = y_{it} | W_i = wi, C_i = ci\right) = \begin{cases} \rho_{ikt} + (1 - \rho_{ikt}) e^{-\lambda_{ikt}}, & y_{it} = 0 \\ (1 - \rho_{ikt}) \frac{\lambda_{ikt}^{y_{it}} e^{-\lambda_{ikt}}}{y_{it}!}, & y_{it} > 0 \end{cases} \tag{3}$$

Loi Censored Normal Model

On considère que la variable Y_{it} est censorée et une variable Y_{it}^* qui suit une loi normale telle que

$$y_{it}^* = f(a_{it}; \beta_k, \delta_k) + \epsilon_{it} = \beta_k A_{it} + \delta_k W_t + \epsilon_{it} \tag{4}$$

où $\epsilon_{it} \sim \mathcal{N}(0;\ \sigma)$
On peut lier y_{it}^* aux données observées et censorées y_{it}.

$$y_{it} = y_{min} \text{ if } y_{it}^* < y_{min} \tag{5}$$

$$y_{it} = y_{it}^* \text{ if } y_{min} \leq y_{it}^* \leq y_{max} \tag{6}$$

$$y_{it} = y_{max} \text{ if } y_{it}^* > y_{max} \tag{7}$$

trajeR

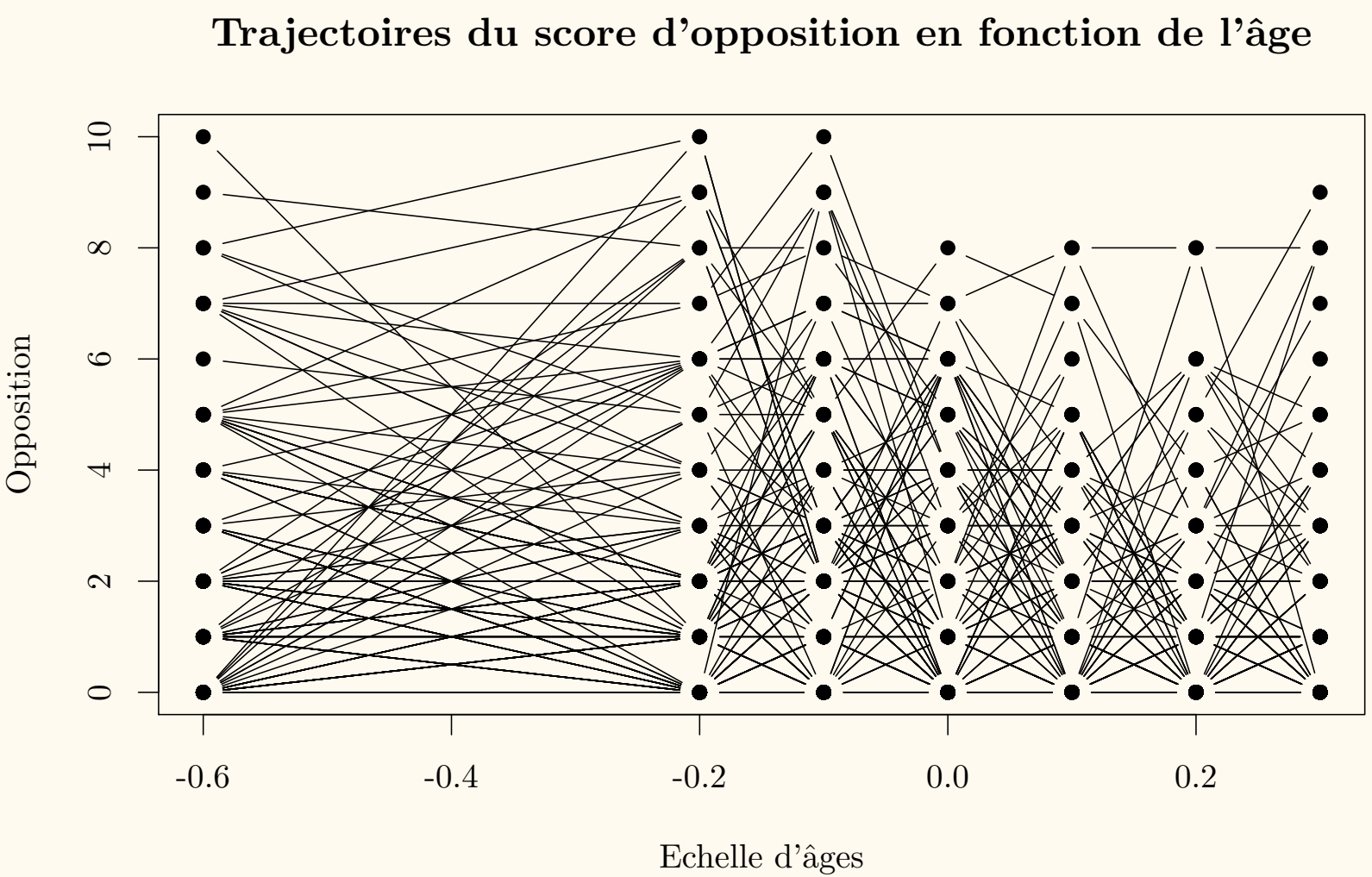
Nagin & Jones ont programmé une porcédure SAS et Stata : `traj`.
On propose un package R

avec 2 méthodes : Likelihood and EM.

Fonction prinicpale :

```
trajeR(Y, A, Risk = NULL, TCOV = NULL, ng, degre, degre.nu
= 0, Model, Method = "L", ssigma = FALSE, ymax = max(Y) +
1, ymin = min(Y) - 1, hessian = TRUE, itermax = 100, paraminit
= NULL, EMIRLS = TRUE, refgr = 1, fct = NULL, diffct =
NULL, nbvar = NULL, nls.lmiter)
```

Exemple illustratif



Exemple illustratif

```
library(trajeR)
```

Exemple illustratif

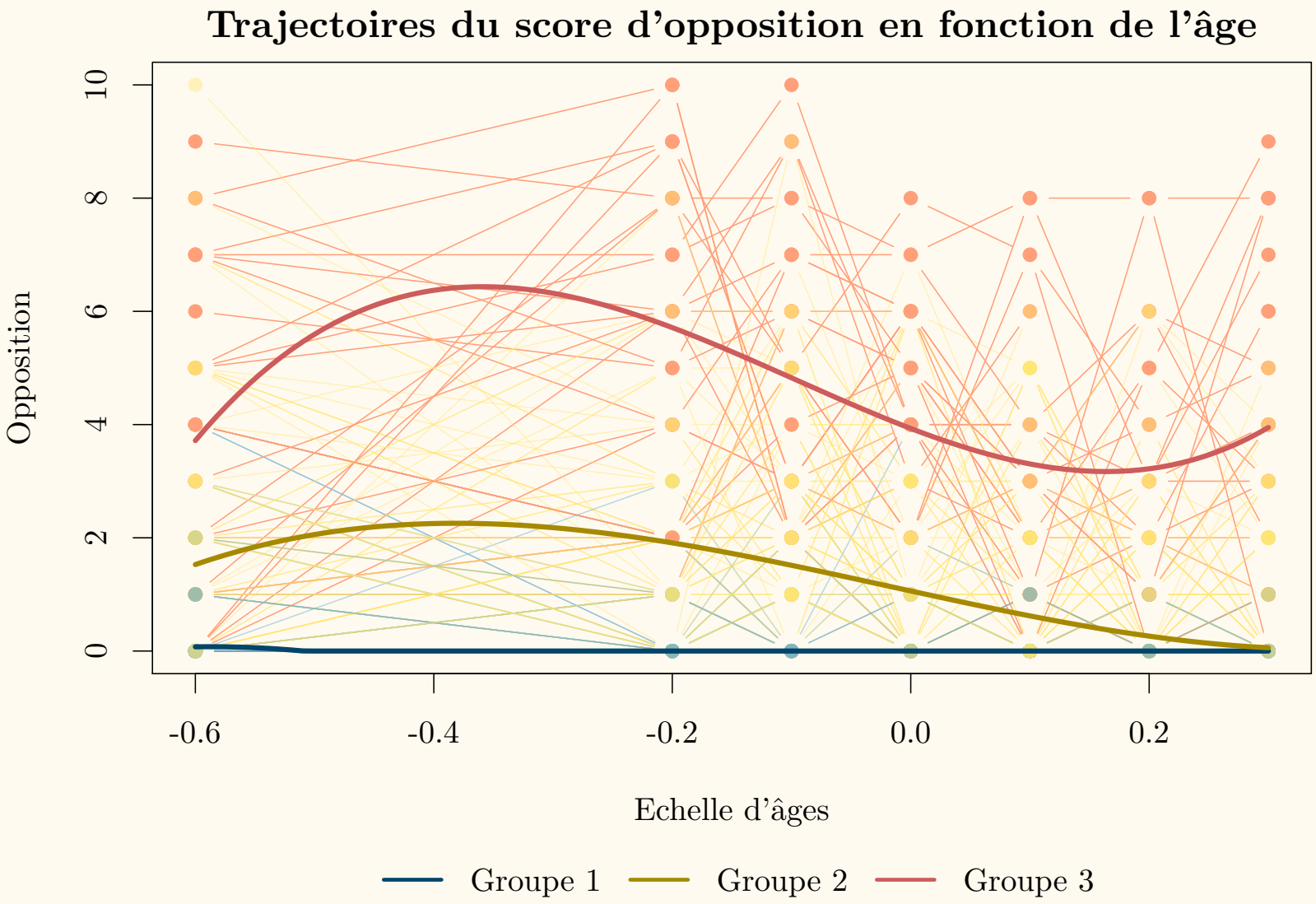
```
library(trajeR)

trajeR(
Y = data[ , 2:8], A = data[ , 9:15],
degre = c(3, 3, 3),
Method = "L", Model = "CNORM",
ymin = 0, ymax = 10,
ssigma = TRUE, hessian = TRUE
)
```

Exemple illustratif

```
Call TrajeR with 3 groups and a 3,3,3 degrees of polynomial shape of trajectory.
Model : Censored Normal
Method : Likelihood
```

group	Parameter	Estimate	Std. Error	T for H0: param.=0	Prob>|T|
1	Intercept	-2.34029	0.59487	-3.93413	9e-05
	Linear	-4.54303	2.97544	-1.52684	0.12713
	Quadratic	5.5429	11.98453	0.4625	0.64382
	Cubic	10.68041	24.10402	0.4431	0.6578
2	Intercept	1.06069	0.3167	3.3492	0.00084
	Linear	-4.58363	1.41108	-3.24831	0.0012
	Quadratic	0.64225	4.71449	0.13623	0.89167
	Cubic	11.64248	10.20263	1.14113	0.2541
3	Intercept	3.93136	0.37738	10.41752	0
	Linear	-8.06229	2.16777	-3.71916	0.00021
	Quadratic	13.36513	6.82152	1.95926	0.05037
	Cubic	45.6647	15.26152	2.99215	0.00284
1	sigma1	2.64271	0.14738	17.93163	0
2	sigma2	2.64271	0.14738	17.93163	0
3	sigma3	2.64271	0.14738	17.93163	0
1	pi1	0.26085	0.07987	0	0.00113
2	pi2	0.54254	0.06186	11.83801	0
3	pi3	0.19661	0.05052	-5.5971	0.00011

Exemple illustratif



Exemple illustratif

On ajoute des variables explicatives pour les groupes : SCOLMER et SCOLPER.

Exemple illustratif

On ajoute des variables explicatives pour les groupes : SCOLMER et SCOLPER.

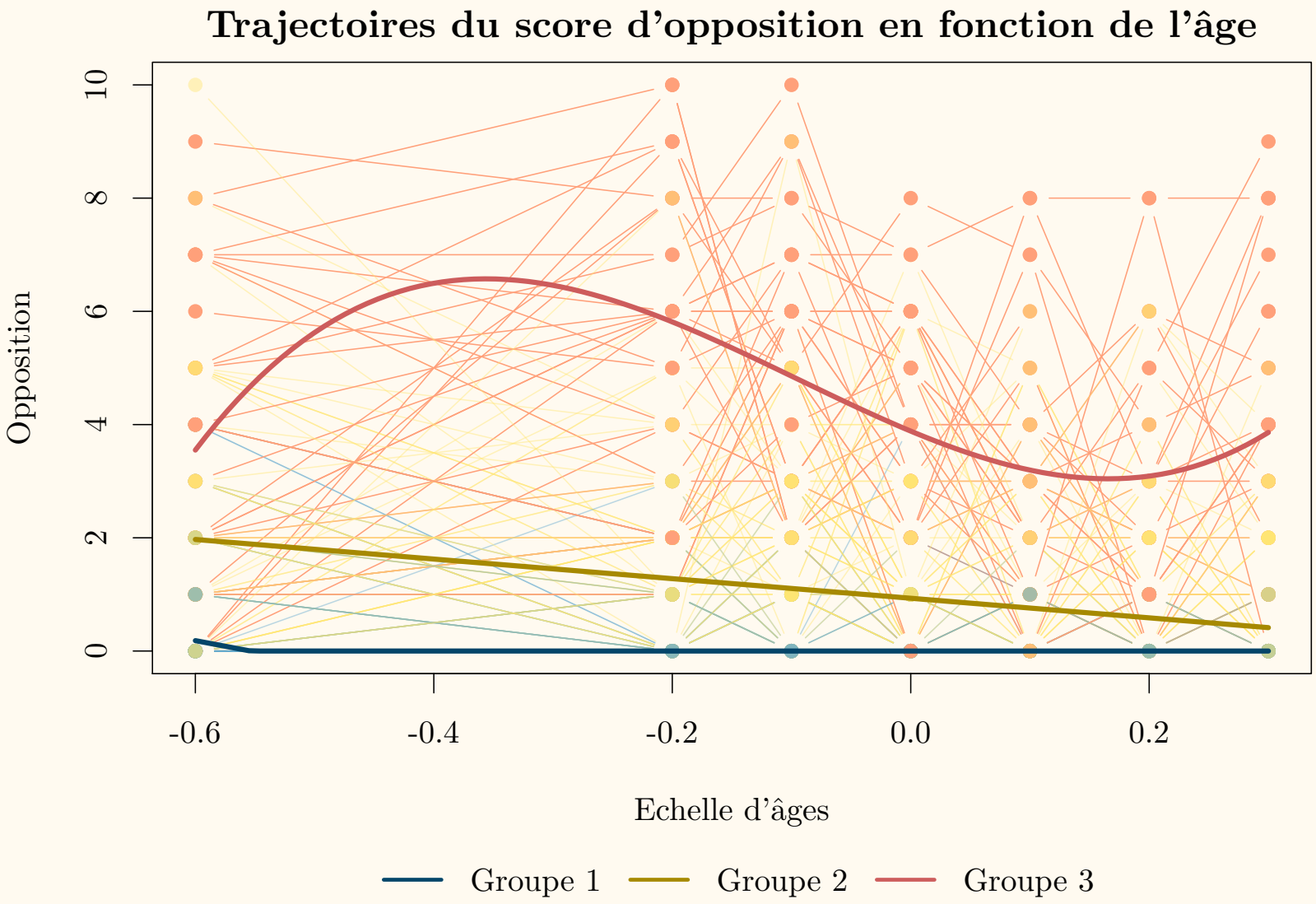
```
trajeR(Y = data[ , 2:8], A = data[ , 9:15], Risk = data[ , 16:17],
                  degre = c(1, 1, 3),
                  Method = "L", Model = "CNORM",
                  ymax = 10, ymin = 0, ssigma = TRUE, hessian = TRUE,
                  paraminit = c(#theta
                    0, 0,0,
                    0.732297366314995,0,0,
                    -0.282750899098949,0,0,
                    #beta
                    -2.34028520478982, -4.54302501861284,
                    1.06068650685758, -4.58363471974226,
                    3.9313576112395, -8.06228539657359, 13.3651329858766,
                    #sigma
                    2.64271037494155, 2.64271037494155, 2.64271037494155)
```

Exemple illustratif

group	Parameter	Estimate	Std. Error	T for H0: param.=0	Prob>|T|
1	Intercept	-2.19652	0.41702	-5.26717	0
	Linear	-3.96666	0.92362	-4.29471	2e-05
2	Intercept	0.93213	0.25762	3.61827	0.00031
	Linear	-1.73025	0.54896	-3.15184	0.00167
3	Intercept	3.88433	0.38775	10.01757	0
	Linear	-8.79596	2.30474	-3.81646	0.00014
	Quadratic	14.18118	7.40042	1.91627	0.05563
	Cubic	49.61899	16.65507	2.97921	0.00296
1	sigma1	2.65235	0.14704	18.03802	0
2	sigma2	2.65235	0.14704	18.03802	0
3	sigma3	2.65235	0.14704	18.03802	0
1	Baseline	0	NA	NA	NA
2	Intercept	3.65064	1.20087	3.04	0.00243
	SCOLMER	-0.05873	0.10072	-0.58305	0.56
	SCOLPER	-0.20242	0.09283	-2.18058	0.02946
3	Intercept	3.82126	1.33147	2.86995	0.0042
	SCOLMER	-0.1606	0.12165	-1.32016	0.18709
	SCOLPER	-0.21765	0.10344	-2.10399	0.03564

Likelihood : -1590.902

Exemple illustratif



trajeR - Fonctions supplémentaires

- Probabilité d'appartenance

```
GroupProb(solCov, Y = data[ , 2:8], A = data[ , 9:15],
          X = data[ , 16:17])
```

```
             [,1]        [,2]         [,3]
[1,] 9.527544e-01 0.047244971 6.070262e-07
[2,] 2.592349e-03 0.994522903 2.884748e-03
[3,] 4.571451e-01 0.542793642 6.130128e-05
[4,] 6.666327e-01 0.333302548 6.473968e-05
[5,] 4.827857e-10 0.004719505 9.952805e-01
[6,] 1.967113e-06 0.067319233 9.326788e-01
```

trajeR - Fonctions supplémentaires

- Probabilité d'appartenance

```
GroupProb(solCov, Y = data[ , 2:8], A = data[ , 9:15],
          X = data[ , 16:17])
```

```
            [,1]        [,2]         [,3]
[1,] 9.527544e-01 0.047244971 6.070262e-07
[2,] 2.592349e-03 0.994522903 2.884748e-03
[3,] 4.571451e-01 0.542793642 6.130128e-05
[4,] 6.666327e-01 0.333302548 6.473968e-05
[5,] 4.827857e-10 0.004719505 9.952805e-01
[6,] 1.967113e-06 0.067319233 9.326788e-01
```

- Profile de groupes

```
GroupProfiles(solCov, Y = data[ , 2:8], A = data[ , 9:15],
          X = data[ , 16:17])
```

```
           Gr 1     Gr 2     Gr 3
SCOLMER 11.44444 10.23684 9.192308
SCOLPER 12.58333 10.30263 9.653846
```

trajeR - Modèle performance

- Probabilité à postériori moyenne

```
AvePP(sol, Y = data[ , 2:8], A = data[ , 9:15])
```

```
[1] 0.8702552 0.8790312 0.8994214
```

```
AvePP(solCov, Y = data[ , 2:8], A = data[ , 9:15],
      X = data[ , 16:17])
```

```
[1] 0.8861916 0.8844400 0.8954563
```

trajeR - Modèle performance

- Probabilité à postériori moyenne

```
AvePP(sol, Y = data[ , 2:8], A = data[ , 9:15])
```

```
[1] 0.8702552 0.8790312 0.8994214
```

```
AvePP(solCov, Y = data[ , 2:8], A = data[ , 9:15],
      X = data[ , 16:17])
```

```
[1] 0.8861916 0.8844400 0.8954563
```

- Ratio de classification correcte

```
OCC(sol, Y = data[ , 2:8], A = data[ , 9:15])
```

```
[1] 19.005929  6.127104 36.541376
```

trajeR - Modèle performance

- Probabilités des groupes estimées contre proportion de l'échantillon assignée à chaque groupe

```
propAssign(sol, Y = data[ , 2:8], A = data[ , 9:15])
```

```
gr
        1         2         3
0.2608696 0.5579710 0.1811594
```

trajeR - Modèle performance

- Probabilités des groupes estimées contre proportion de l'échantillon assignée à chaque groupe

```
propAssign(sol, Y = data[ , 2:8], A = data[ , 9:15])
```

```
gr
        1         2         3
0.2608696 0.5579710 0.1811594
```

- Intervalle de confiance

```
ConfIntT(sol, Y = data[ , 2:8], A = data[ , 9:15],
          nb = 10000, alpha = 0.98)
```

```
         [,1]      [,2]      [,3]
1%  0.2212523 0.4964141 0.1725465
99% 0.3039079 0.5871770 0.2236614
```

trajeR - Modèle performance

- Résumé

```
adequacy(sol, Y = data[ , 2:8], A = data[ , 9:15],
         nb = 10000, alpha = 0.98)
```

```
                    1         2          3
Prob. est.  0.2608541 0.5425382  0.1966077
CI inf.     0.2213079 0.4961152  0.1714537
CI sup.     0.3045968 0.5871363  0.2233511
Prop.       0.2608696 0.5579710  0.1811594
AvePP       0.8702552 0.8790312  0.8994214
OCC        19.0059288 6.1271043 36.5413764
```

trajeR - Modèle sélection

- AIC – `trajeRAIC(...)`
- BIC – `trajeRBIC(...)`
- Slope Heuristics – `trajeRSH(...)`

References

Jones, Bobby L. and Daniel S. Nagin (Nov. 2013). "A Note on a Stata Plugin for Estimating Group-based Trajectory Models". In: *Sociological Methods & Research* 42.4, pp. 608–613. ISSN: 0049-1241, 1552-8294. DOI: 10.1177/0049124113503141.

Nagin, Daniel (2005). *Group-based modeling of development*. Cambridge, Mass: Harvard University Press. ISBN: 978-0-674-01686-6.

Nagin, Daniel S. and Richard E. Tremblay (Nov. 2005). "Developmental trajectory groups: fact or a useful statistical fiction?" In: *Criminology* 43.4, pp. 873–904. ISSN: 0011-1384, 1745-9125. DOI: 10.1111/j.1745-9125.2005.00026.x.

Noel, Cédric and Jang Schiltz (2021). *TrajeR - an R package for finite mixture models.* SMTDA 2020.

Schiltz, Jang (2015). "A Generalization of Nagin's Finite Mixture Model". In: *Dependent Data in Social Sciences Research*. Ed. by Mark Stemmler, Alexander von Eye, and Wolfgang Wiedermann. Cham: Springer International Publishing, pp. 107–123. ISBN: 978-3-319-20585-4.