PAGANZ 2024 Pumas Workshop

Mohamed Tarek

Senior Product Engineer at PumasAI Inc.

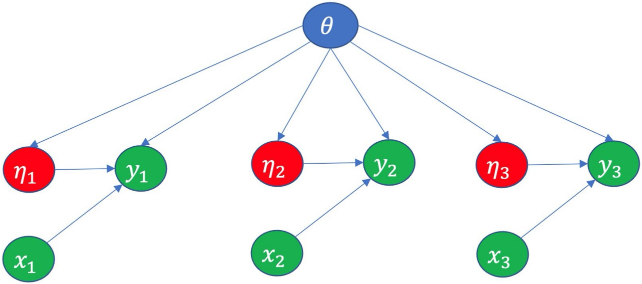
Research Affiliate at University of Sydney Business School

Contents I

NLME fitting algorithms

NLME notation

Assume there are 3 subjects



PumasAI

NLME notation

```
@model begin
```

θ

```
  @param begin
    θ ∈ VectorDomain(4, lower = zeros(4))
    Ω ∈ PSDDomain(2)
    Σ ∈ RealDomain(lower = 0.0)
    a ∈ RealDomain(lower = 0.0, upper = 1.0)
  end
```

$\eta_i \mid \theta$

```
  @random begin
    η ~ MvNormal(Ω)
  end
```

x_i

```
  @covariates sex wt etn
```

$y_i \mid \theta, \eta_i, x_i$

```
  @pre begin
    θ1 := θ[1]
    Ka = θ1
    CL = θ[2] * ((wt / 70)^0.75) * (θ[4]^sex) *
      exp(η[1])
    Vc = θ[3] * exp(η[2])
  end
  @dynamics begin
    Depot' = -Ka * Depot
    Central' = Ka * Depot - (CL / Vc) * Central
    Res' = Depot - Central
  end
  @derived begin
    conc = @. Central / Vc
    dv ~ @. Normal(conc, conc * Σ)
```

```
    T_max = maximum(t)
  end
  @observed begin
    obs_cmax = maximum(dv)
  end
end
```

NLME notation

- θ: all the population parameters (typical values, BSV, BOV and residual variability)

NLME notation

- θ: all the population parameters (typical values, BSV, BOV and residual variability)
- η_i: random effects of subject i

NLME notation

- θ: all the population parameters (typical values, BSV, BOV and residual variability)
- η_i: random effects of subject i
- N: number of subjects

NLME notation

- θ: all the population parameters (typical values, BSV, BOV and residual variability)
- η_i: random effects of subject i
- N: number of subjects
- $\eta = \{\eta_i, \forall i \in 1 \ldots N\}$: random effects of all the subjects

NLME notation

- θ: all the population parameters (typical values, BSV, BOV and residual variability)
- η_i: random effects of subject i
- N: number of subjects
- $\eta = \{\eta_i, \forall i \in 1 \ldots N\}$: random effects of all the subjects
- y_i: observations of subject i

PumasAI

NLME notation

- θ: all the population parameters (typical values, BSV, BOV and residual variability)
- η_i: random effects of subject i
- N: number of subjects
- $\eta = \{\eta_i, \forall i \in 1 \ldots N\}$: random effects of all the subjects
- y_i: observations of subject i
- x_i: covariates of subject i

NLME notation

- $p(y = y_i \mid \theta = \theta, \eta = \eta_i, x = x_i) = p(y_i \mid \theta, \eta_i, x_i)$: likelihood of (θ, η_i) given subject i's data (x_i, y_i). Also known as the conditional probability of y_i given θ, η_i, x_i. Or just the conditional likelihood of η_i given θ and (x_i, y_i).

PumasAI

NLME notation

- $p(y = y_i \mid \theta = \theta, \eta = \eta_i, x = x_i) = p(y_i \mid \theta, \eta_i, x_i)$: likelihood of (θ, η_i) given subject i's data (x_i, y_i). Also known as the conditional probability of y_i given θ, η_i, x_i. Or just the conditional likelihood of η_i given θ and (x_i, y_i).
- $p(\eta = \eta_i \mid \theta = \theta) = p(\eta_i \mid \theta)$: prior probability of the random effects η_i given the population parameters θ.

PumasAI

NLME notation

- $p(y = y_i \mid \theta = \theta, \eta = \eta_i, x = x_i) = p(y_i \mid \theta, \eta_i, x_i)$: likelihood of (θ, η_i) given subject i's data (x_i, y_i). Also known as the conditional probability of y_i given θ, η_i, x_i. Or just the conditional likelihood of η_i given θ and (x_i, y_i).
- $p(\eta = \eta_i \mid \theta = \theta) = p(\eta_i \mid \theta)$: prior probability of the random effects η_i given the population parameters θ.
- $p(y = y_i \mid \theta = \theta, x = x_i) = p(y_i \mid \theta, x_i) = \int p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta) \, d\eta_i$: marginal likelihood of θ given subject i's data y_i.

PumasAI

Marginal likelihood maximization

$$\theta^* = \arg max_\theta \prod_{i=1}^{N} p(y_i \mid \theta, x_i)$$

$$= \arg max_\theta \prod_{i=1}^{N} \int p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta) \, d\eta_i$$

$$\text{EBE}_i = \eta_i^* = \arg max_{\eta_i} \left(p(y_i \mid \theta = \theta^*, \eta_i, x_i) \cdot p(\eta_i \mid \theta = \theta^*) \right)$$

General Laplace method

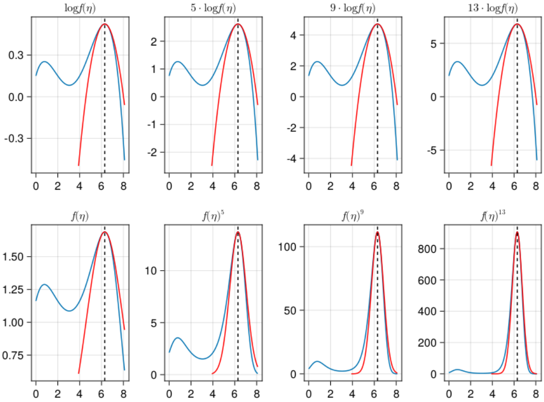
$$\int f(\eta)\, d\eta \approx f(\eta^*)\sqrt{(2\pi)^m/|-H|}$$

- f: a positive scalar-valued functon of η
- η: vector of m integration variables
- η^*: global maximizer of $\log f$, $\frac{d \log f}{d\eta}(\eta^*) = 0$
- H: second derivative matrix of $\log f$ wrt η at η^*, must be negative definite at $\eta = \eta^*$
- $|-H|$: determinant of $-H$

PumasAI

NLME Laplace method

Laplace uses a second order Taylor series approximation of $\log f$ at η^*.



PumasAI

NLME Laplace method

Consider the 2 local maximizers η_1 (lower peak) and η_2 (higher peak).

$$\begin{aligned} c &= \log f(\eta_2) - \log f(\eta_1) \\ n \cdot c &= n \cdot (\log f(\eta_2) - \log f(\eta_1)) \\ e^{n \cdot c} &= f(\eta_2)^n / f(\eta_1)^n \end{aligned}$$

Summary

Approximation error of $n \cdot \log f$ away from the mode η^* is not significant as n increases.

NLME Laplace method

- There are N functions $\{f_i : i = 1 \ldots N\}$ to be integrated, one for each subject
- $f_i(\eta_i) = p(y_i \mid \theta, \eta_i, x_i) \cdot p(\eta_i \mid \theta)$
- Laplace method

$$\int f_i(\eta_i)\, d\eta_i = f_i(\text{EBE}_i)\sqrt{(2\pi)^m / |-H_i|}$$

- H_i: second derivative matrix of $\log f_i$ wrt η_i at EBE_i, must be negative definite at $\eta_i = \text{EBE}_i$

PumasAI

General FOCE(I)

FOCE(I) approximates the Hessian H_i for each subject i. Assume the following:

$$\begin{aligned} \log f_i(\eta_i) &= \log p(y_i \mid \theta, \eta_i, x_i) + \log p(\eta_i \mid \theta) \\ &= L_i(g_i(\eta_i)) + \log p(\eta_i \mid \theta) \end{aligned}$$

where:

- g_i returns the vector of IPREDs μ_i and the residual standard deviations σ_i (constant in the additive error model case), at all observed time points, and
- L_i is the log probability of y_i given μ_i and σ_i

General FOCE(I)

g_i is usually the most expensive component of $\log f_i$, because it often involves solving a differential equation. So let's approximate it!

First order Taylor series approximation

- FO

$$g_i(\eta_i) \approx g_i(0) + \frac{dg_i}{d\eta_i}(0) \cdot \eta_i$$

- FOCE(I)

$$g_i(\eta_i) \approx g_i(\mathsf{EBE}_i) + \frac{dg_i}{d\eta_i}(\mathsf{EBE}_i) \cdot (\eta_i - \mathsf{EBE}_i)$$

PumasAI

General FOCE(I)

Summary

- FOCE(I) ensures that the approximation error in g_i (and $\log f_i$ by extension) is low in the proximity of EBE_i.
- FO does not ensure that so it only works well if:
  - EBE_i is not far from 0, or
  - g_i is close to linear in the interval $[0, \text{EBE}_i]$.
- FO requires a correction term in the Laplace method because the gradient of $\log f_i$ wrt η_i at $\eta_i = 0$ is not 0.

PumasAI

General FOCE(I)

Chain rule for Hessians

$$(L_i \cdot g_i)''(\eta_i) = \frac{dg_i}{d\eta_i}^T \cdot \frac{\partial^2 L_i}{\partial g_i \cdot \partial g_i^T} \cdot \frac{dg_i}{d\eta_i} + \sum_{t=1}^{d} \left(\frac{\partial L_i}{\partial g_{i,t}} \cdot \underbrace{\frac{\partial^2 g_{i,t}}{\partial \eta_i \cdot \partial \eta_i^T}}_{\text{0 if linear}} \right)$$

where d is twice the number of observed time points (corresponding to μ_i and σ_i) and $g_{i,t}$ is the t^{th} component of g_i.

PumasAI

General FOCE(I)

Summary

If g_i is linear in η_i:

- $\frac{\partial^2 g_{i,t}}{\partial \eta_i \cdot \partial \eta_i^T} = 0$
- $J_i = \frac{\partial g_i}{\partial \eta_i}$ is constant

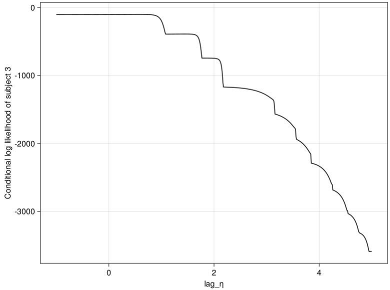
The Hessian simplifies to:

$$(L_i \cdot g_i)''(\eta_i) = J_i^T \cdot \frac{\partial^2 L_i}{\partial g_i \cdot \partial g_i^T} \cdot J_i$$

PumasAI

General FOCE(I)

- One surprising advantage of FOCE(I) is that the Hessian approximation is often negative definite even when the exact Hessian is singular or not well defined at $\eta_i = \eta_i^*$.



PumasAI

General FOCE(I)

- J_i can be computed for each subject using finite difference at
  - $\eta_i = 0$ for FO, or
  - $\eta_i = \text{EBE}_i$ for FOCE(I)
- For many data distributions, $\frac{\partial^2 L_i}{\partial g_i \cdot \partial g_i^T}$ is both diagonal and has a closed form. Doesn't have to be Gaussian!

PumasAI

General FOCE(I)

Recall

$$\begin{aligned}\log f_i(\eta_i) &= \log p(y_i \mid \theta, \eta_i, x_i) + \log p(\eta_i \mid \theta) \\ &= L_i(\underbrace{g_i(\eta_i)}_{\text{approx}}) + \log p(\eta_i \mid \theta)\end{aligned}$$

For many random effects distributions, the Hessian of $\log p(\eta_i \mid \theta)$ wrt η_i has a closed form. Doesn't have to be Gaussian!

PumasAI

General FOCE(I)

- Pumas FOCE supports a number of data distributions:
  - **Continuous**: Normal, LogNormal, Gamma, Exponential, Beta
  - **Discrete**: NegativeBinomial, Bernoulli, Binomial, Poisson, Categorical

PumasAI

General FOCE(I)

- Pumas supports a number of random effect distributions:
  - **Unbounded**: Cauchy, Gumbel, Laplace, Logistic, Normal, NormalCanon, NormalInverseGaussian, PGeneralizedGaussian, TDist
  - **Positive**: BetaPrime, Chi, Chisq, Erlang, Exponential, Frechet, Gamma, InverseGamma, InverseGaussian, Kolmogorov, LogNormal, NoncentralChisq, Rayleigh, Weibull
  - **Between 0 and 1**: Beta, LogitNormal
  - **Other bounded**: Uniform, Arcsine, Biweight, Cosine, Epanechnikov, LogUniform, Semicircle, SymTriangularDist, Triweight

PumasAI

General FOCE(I)

Summary

- In Pumas, FOCE is always "with interaction".
- Use FOCE if supported, otherwise use Laplace.
- Avoid FO.

Weighted residuals

Distribution of response

Assuming Gaussian random effects and error model, for each subject i, the conditional distribution $p(y_i \mid x_i, \theta)$ is given by:

$$\begin{aligned} \eta_i &\sim \mathcal{N}(0, \Omega) \\ (\mu_i, \sigma_i) &= g(\eta_i; x_i) \\ y_i &\sim \mathcal{N}(\mu_i, \sigma_i) \end{aligned}$$

where $g(\eta_i; x_i) = g_i(\eta_i)$ (same functional form g for all subjects).

PumasAI

Distribution of response

Alternative representation

$$
\begin{aligned}
\eta_i &\sim \mathcal{N}(0, \Omega) \\
(\mu_i, \sigma_i) &= g(\eta_i; x_i) \\
\epsilon_{i,t} &\sim \mathcal{N}(0, 1) \\
y_{i,t} &= \mu_{i,t} + \sigma_{i,t} \cdot \epsilon_{i,t}
\end{aligned}
$$

where the t is the index for the number of observations per subject.

Distribution of response

The machine learning (ML) community call this class of models:

- (Conditional) generative models, or
- Latent variable models

Congratulations, you have been doing ML this whole time!

Distribution of response

- $p(y_i \mid x_i, \theta)$ is the distribution we sample from when doing a visual predictive check (VPC) to compare the distribtuion of simulated y_i to the distribution of observed y_i.
- The weighted residual is

$$\text{WRES}_{i,t} = \frac{y_{i,t} - E[y_{i,t} \mid x_i]}{\sqrt{Var[y_{i,t} \mid x_i]}} \sim \mathcal{N}(0, 1)$$

- **Problem**: $p(y_i \mid x_i, \theta)$ (in general) has no closed form mean and variance.
- **Solution**: let's approximate it!

Approximate distribution of response

First order Taylor series approximation

- FO

$$\mu_i(\eta_i) \approx \mu_i(0) + \frac{d\mu_i}{d\eta_i}(0) \cdot \eta_i$$
$$\sigma_i(\eta_i) \approx \sigma_i(0) + \frac{d\sigma_i}{d\eta_i}(0) \cdot \eta_i$$

- FOCEI

$$\mu_i(\eta_i) \approx \mu_i(\mathrm{EBE}_i) + \frac{d\mu_i}{d\eta_i}(\mathrm{EBE}_i) \cdot (\eta_i - \mathrm{EBE}_i)$$
$$\sigma_i(\eta_i) \approx \sigma_i(\mathrm{EBE}_i) + \underbrace{\frac{d\sigma_i}{d\eta_i}(\mathrm{EBE}_i)}_{\neq 0 \text{ in general}} \cdot (\eta_i - \mathrm{EBE}_i)$$

PumasAI

Approximate distribution of response

Approximate means

- FO

$$E[\mu_i] \approx \mu_i(0)$$
$$E[\sigma_i] \approx \sigma_i(0)$$

- FOCEI

$$E[\mu_i] \approx \mu_i(\mathrm{EBE}_i) - \frac{d\mu_i}{d\eta_i}(\mathrm{EBE}_i) \cdot \mathrm{EBE}_i$$
$$E[\sigma_i] \approx \sigma_i(\mathrm{EBE}_i) - \frac{d\sigma_i}{d\eta_i}(\mathrm{EBE}_i) \cdot \mathrm{EBE}_i$$

Approximate distribution of response

Approximate variances

- FO

$$Var[\mu_i] \approx \frac{d\mu_i}{d\eta_i}(0) \cdot \Omega \cdot \frac{d\mu_i}{d\eta_i}(0)^T$$

$$Var[\sigma_i] \approx \frac{d\sigma_i}{d\eta_i}(0) \cdot \Omega \cdot \frac{d\sigma_i}{d\eta_i}(0)^T$$

- FOCEI

$$Var[\mu_i] \approx \frac{d\mu_i}{d\eta_i}(\mathsf{EBE}_i) \cdot \Omega \cdot \frac{d\mu_i}{d\eta_i}(\mathsf{EBE}_i)^T$$

$$Var[\sigma_i] \approx \frac{d\sigma_i}{d\eta_i}(\mathsf{EBE}_i) \cdot \Omega \cdot \frac{d\sigma_i}{d\eta_i}(\mathsf{EBE}_i)^T$$

Approximate distribution of response

Recall

$$y_{i,t} = \mu_{i,t} + \sigma_{i,t} \cdot \epsilon_{i,t}$$

Mean

$$\begin{aligned} E[y_{i,t} \mid x_i] &= E[\mu_{i,t}] + E[\sigma_{i,t}] \cdot \overbrace{E[\epsilon_{i,t}]}^{0} \\ &= E[\mu_{i,t}] \end{aligned}$$

PumasAI

Approximate distribution of response

Recall

$$y_{i,t} = \mu_{i,t} + \sigma_{i,t} \cdot \epsilon_{i,t}$$

Variance

$$\begin{aligned} Var[y_{i,t} \mid x_i] &= Var[\mu_{i,t}] + Var[\sigma_{i,t} \cdot \epsilon_{i,t}] - \underbrace{Cov[\mu_{i,t}, \sigma_{i,t} \cdot \epsilon_{i,t}]}_{0} \\ &= Var[\mu_{i,t}] + Var[\sigma_{i,t} \cdot \epsilon_{i,t}] \\ &= Var[\mu_{i,t}] + Var[\sigma_{i,t}] + E[\sigma_{i,t}]^2 \end{aligned}$$

PumasAI

Approximate distribution of response

$$Cov[\mu_{i,t}, \sigma_{i,t} \cdot \epsilon_{i,t}] = \overbrace{E[\mu_{i,t} \cdot \sigma_{i,t} \cdot \epsilon_{i,t}]}^{0} - E[\mu_{i,t}] \cdot \overbrace{E[\sigma_{i,t} \cdot \epsilon_{i,t}]}^{0}$$

$$E[\mu_{i,t} \cdot \sigma_{i,t} \cdot \epsilon_{i,t}] = E[\mu_{i,t} \cdot \sigma_{i,t}] \cdot \overbrace{E[\epsilon_{i,t}]}^{0} = 0$$

$$E[\sigma_{i,t} \cdot \epsilon_{i,t}] = E[\sigma_{i,t}] \cdot \overbrace{E[\epsilon_{i,t}]}^{0} = 0$$

PumasAI

Approximate distribution of response

Summary

After the FO/FOCEI approximation, we were able to obtain closed form approximations of $E[y_{i,t} \mid x_i]$ and $Var[y_{i,t} \mid x_i]$.

$$\text{WRES}_{i,t} = \frac{y_{i,t} - E[y_{i,t} \mid x_i]}{\sqrt{Var[y_{i,t} \mid x_i]}}$$

Standard error estimation

Goal

Estimate the covariance matrix of the estimator θ^*:

$$\theta^* = \arg max_\theta \prod_{i=1}^{N} p(y_i \mid \theta, x_i)$$

Sandwich estimator of standard errors

Asymptotic covariance

$$\theta^* \sim \mathcal{N}(\theta_0, V) = \mathcal{N}(\theta_0, A^{-1}BA^{-1})$$

$$A = \sum_{i=1}^{N} \frac{\partial^2 \log p(y_i \mid \theta, x_i)}{\partial\theta \cdot \partial\theta^T}(\theta_0)$$

$$B = \sum_{i=1}^{N} \frac{\partial \log p(y_i \mid \theta, x_i)}{\partial\theta}(\theta_0) \times \frac{\partial \log p(y_i \mid \theta, x_i)}{\partial\theta}(\theta_0)^T$$

where θ_0 is the set of unknown true parameters.

PumasAI

Sandwich estimator of standard errors

Estimated covariance

$$\theta^* \overset{a}{\sim} \mathcal{N}(\theta_0, \hat{V}) = \mathcal{N}(\theta_0, \hat{A}^{-1}\hat{B}\hat{A}^{-1})$$

$$\hat{A} = \sum_{i=1}^{N} \frac{\partial^2 \log p(y_i \mid \theta, x_i)}{\partial\theta \cdot \partial\theta^T}(\theta^*)$$

$$\hat{B} = \sum_{i=1}^{N} \frac{\partial \log p(y_i \mid \theta, x_i)}{\partial\theta}(\theta^*) \times \frac{\partial \log p(y_i \mid \theta, x_i)}{\partial\theta}(\theta^*)^T$$

PumasAI

Sandwich estimator of standard errors

Standard error estiamtes

The square root of the diagonal elements of $\hat{V}$ are the standard error estimates of θ^*.

Sandwich estimator of standard errors

Computing $\hat{V}$

- $\hat{A}$ and $\hat{B}$ can be approximated with finite difference
- $\log p(y_i \mid \theta, x_i)$ itself needs to be approximated with Laplace/FOCE/FO
- $\hat{A}^{-1} \cdot \hat{B} \cdot \hat{A}^{-1}$ is computed using a generalized eigenvalue problem.

PumasAI

Sandwich estimator of standard errors

Computing $\hat{V}$

Consider the following generalized eigenvalue problem:

$$\hat{B} \cdot U = \hat{A} \cdot U \cdot \Lambda$$
$$I = U^T \cdot \hat{A} \cdot U$$

where U is the matrix of generalized eigenvectors and Λ is the diagonal matrix of generalized eigenvalues.

Sandwich estimator of standard errors

Computing $\hat{V}$

The inverse of the matrix of eigenvectors U is obtained from the following constraint on U:

$$(U^T \cdot \hat{A}) \cdot U = I$$
$$U^{-1} = U^T \cdot \hat{A}$$

PumasAI

Sandwich estimator of standard errors

Computing $\hat{V}$

The following identity is true:

$$\hat{B} \cdot U = \hat{A} \cdot U \cdot \Lambda$$
$$\hat{A}^{-1} \cdot \hat{B} \cdot U = U \cdot \Lambda$$
$$\hat{A}^{-1} \cdot \hat{B} = U \cdot \Lambda \cdot U^{-1}$$
$$\hat{A}^{-1} \cdot \hat{B} = U \cdot \Lambda \cdot U^{T} \cdot \hat{A}$$
$$\hat{V} = \hat{A}^{-1} \cdot \hat{B} \cdot \hat{A}^{-1} = U \cdot \Lambda \cdot U^{T}$$

Sandwich estimator of standard errors

Failed estimator

- If the computed $\hat{A}$ is: a) singular, b) near singular, or c) has negative eigenvalues, the sandwich estimator will fail.
- This is a sign of poor identifiability of at least 1 parameter and/or significant numerical errors.
- Even if a single (IIV) parameter is not identifiable given the data, $\hat{A}$ will be singular.

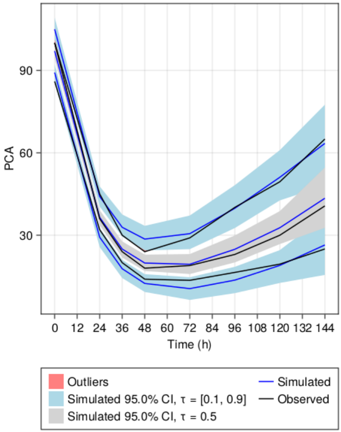
Sandwich estimator of standard errors

Failed estimator

- Numerical errors in the finite difference or Laplace/FOCE/FO can also cause the computed approximate $\hat{A}$ to be singular (or have small negative eigenvalues) even when the exact matrix $\hat{A}$ may be only *near* singular and positive definite.

Continuous visual predictive check

Example


90
60
30
PCA
0
12
24
36
48
60
72
84
96
108
120
132
144
Time (h)
Outliers
Simulated 95.0% CI, τ = [0.1, 0.9]
Simulated 95.0% CI, τ = 0.5
Simulated
Observed

Continuous VPC procedure

1. Simulate a synthetic population a given number of samples (`samples`, default 499).
2. Stratify the observed and simulated populations by the stratification variable.
3. For each simulated population stratum, do smoothed quantile regression at `nnodes` nodes picked from the the data.
   - Default quantiles: 0.1, 0.5 and 0.9.
   - Default `nnodes`: 11
   - Default smoothing `bandwidth`: 2.0

Continuous VPC procedure

4. Find the (hyper-)quantiles of the per-scenario population quantiles within each stratum,
   - Hyper-quantiles:
     - (1 - `level`) / 2
     - 0.5 (`simquantile_medians` hidden by default)
     - (1 + `level`) / 2
   - Default `level`: 0.95
5. For each observed population stratum, repeat step 3.
6. For each stratum, plot the population's quantiles and the hyper-quantiles of each simulated quantile.

PumasAI

Time to event models

Definitions

- Instantaneous hazard

$$\lambda(t) > 0$$

- Cumulative hazard

$$\Lambda(t) = \int_0^t \lambda(t')\,dt'$$

- Survival function: probability of survival up to time t

$$S(t) = \exp(-\Lambda(t))$$

Definitions

- Failure function: probability of death/failure before time t

$$F(t) = 1 - S(t)$$

- Probability density function of time of death t

$$f(t) = \frac{dF}{dt} = \lambda(t) \cdot \exp(-\Lambda(t))$$

- Expected time of death $E[t]$

$$E[t] = \int_0^{\infty} t \cdot f(t)\, dt = \int_0^{\infty} S(t)\, dt$$

Log likelihood

The log likelihood for censored survival data is given by the following 2 formulas:

- For censored subjects at time t (patient survived until time t)

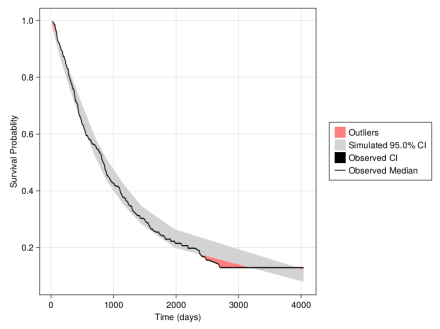
$$\log \text{likelihood} = \log S(t) = -\Lambda(t)$$

- For subjects dead at time t

$$\log \text{likelihood} = \log f(t) = \log \lambda(t) - \Lambda(t)$$

Time to event VPC procedure

Example



Time to event VPC procedure

1. Simulate a synthetic population a given number of samples (`samples`, default 499). For each subject:
   1. Evaluate the cumulative hazard function Λ at `nT` (default 10) time points between `minT` and `maxT`.
   2. Use a cubic spline to interpolate between the Λ values.
   3. Use inverse CDF transform sampling to sample the time of death from the cumulative hazard function.

PumasAI

Time to event VPC procedure

2. Stratify the observed and simulated populations by the stratification variable.
3. For each simulated population stratum:
   1. Estimate the Kaplan Meier (KM) curve. d_i is the number of deaths at t_i and n_i is the number of people at risk at time t_i.

$$\hat{S}(t) = \prod_{i:t_i<t} \left(1 - \frac{d_i}{n_i}\right)$$

   2. Combine all simulated populations' KM curves into one data frame.
   3. Do quantile regression with smoothing to get smooth curves for the quantiles at a number of nodes `nnodes` (default 11).

Time to event VPC procedure

4. For each observed population stratum, estimate the KM curve.
5. Plot the observed KM curve against the smoothed quantiles for each stratum.

Time to event VPC procedure

Inverse CDF sampling

- If $R \sim$ Uniform$(0, 1)$, then $-\log(1 - R) \sim$ Exponential(1).

$$\begin{aligned} F(t) &\leq R \\ 1 - S(t) &\leq R \\ \exp(-\Lambda(t)) &\geq 1 - R \\ \Lambda(t) &\leq -\log(1 - R) \end{aligned}$$

- The sample t is obtained using a root finding algorithm to find the root for $\Lambda(t) = -\log(1 - R)$.

PumasAI