LSTAT2020 : Calcul Statistique sur Ordinateur
Projet R - 2013/2014

Consignes et calendrier

Ce travail a pour objectif de tester vos compétences acquises en R au travers des 3 séances d'exercices. Il est à réaliser par groupe de 1 à 2 personnes et compte pour 3 points (/20) à l'examen.
Chaque étudiant devra s'inscrire dans un groupe sur iCampus. Chaque groupe pourra alors poster sur iCampus le travail réalisé, **pour le lundi 25 novembre 2013 à 16h00 au plus tard**. Pour cela, dans la rubrique « Travaux », sélectionnez le groupe où vous êtes inscrits et choisissez l'option « **nouvelle soumission** ». **ATTENTION, ne déposez pas votre travail dans la rubrique « Documents » de votre groupe. Les travaux qui ne seront pas soumis correctement ne seront pas corrigés.**

Format et description de la soumission

1. Vous posterez un dossier compressé (.zip) contenant :
   - Le(s) script(s) R.
   - Les documents générés par votre code.
2. Vous nommerez votre dossier zip de la manière suivante :
   SoumissionFinale_Nom1_Nom2.

   ATTENTION, les travaux qui ne seront pas postés en format compressé ne seront pas corrigés. Afin d'obtenir un note pour le projet, le nom de tous les étudiants du groupe doit IMPERATIVEMENT se trouver dans le nom du dossier compressé.

Les TP's n'ont pas lieu durant la semaine du 12 novembre mais la salle informatique habituelle est à votre disposition pour préparer votre travail durant les heures normales de TP. Vincent Bremhorst sera disponible à ces moments dans la salle informatique pour répondre à vos éventuelles questions. Dans la réalisation de votre travail, il est strictement INTERDIT de fournir une partie de votre code à un autre groupe. Si nous constatons que des groupes ont des programmes semblables, des mesures appropriées seront prises. Notez qu'il y a par ailleurs de multiples façons d'arriver au résultat demandé.

Contexte et description du travail

Lors de ce projet, nous vous demanderons d'analyser une base de données contenant des données de survie. Ces données seront analysées à l'aide d'un modèle de survie appelé **Modèle de**

Cox. Les paramètres de ce modèle seront estimés en utilisant **l'algorithme de Metropolis**. Ci-dessous se trouve une introduction aux différents concepts à utiliser lors du projet.

Introduction à l'analyse de survie et au modèle de Cox

L'analyse de survie est une branche des statistiques qui étudie le temps nécessaire à la réalisation d'un évènement d'intérêt, par exemple le temps de réapparition d'une maladie en médecine, ou, en sociologie, le temps de récidive d'un ancien condamné après sa sortie de prison. La réalisation du phénomène étudié est appelée un événement. Cependant, il est possible que certains sujets présents dans l'étude quittent celle-ci avant de subir le phénomène d'intérêt. Dès lors, la seule information dont nous disposons est qu'à leur sortie de l'étude, ces sujets n'avaient pas encore subi l'évènement d'intérêt (leur temps d'événement est supérieur au temps observé). On dira d'eux qu'ils sont censurés à droite.
Voici les définitions des fonctions propres à l'analyse de survie dont vous aurez besoin pour la réalisation du projet :

Soit la variable aléatoire T, représentant la réalisation de l'évènement d'intérêt. Supposons que la distribution de T soit définie par la fonction de Répartition $F^T(t) = P(T \leq t)$ et de fonction de densité $f^T(t)$, nous avons :

i) La fonction de survie $S(t) = P(T \geq t) = 1 - F(t)$ représente la probabilité qu'à l'instant t, le sujet n'ait pas encore subi l'évènement d'intérêt.

ii) La fonction de hasard $h(t)$ correspond au risque d'observer l'évènement de manière instantanée dans $[t, t + dt]$ sachant que celui-ci n'a pas encore été observé à l'instant t. On peut l'obtenir de la manière suivante : $h(t) = \frac{f(t)}{S(t)}$.

iii) δ est appelé indicateur de censure. $\delta = 1$ si l'évènement a été observé et $\delta = 0$ si le sujet est censuré à droite.

iv) La fonction de vraisemblance des données de survie pourvues de censure à droite est définie par :

$$Lik = \prod_{i=1}^{n} h(t_i)^{\delta_i} S(t_i)$$

Le modèle de Cox permet de modéliser la fonction de hasard $h(t)$ ou la fonction de survie $S(t)$ et est défini de la manière suivante :

$$\mathrm{h(t)} = \mathrm{h_0}(t) \exp(\boldsymbol{x}^T \boldsymbol{\beta}) \quad \text{ou} \quad \mathrm{S(t)} = \mathrm{S_0}(t)^{\exp(\boldsymbol{x}^T \boldsymbol{\beta})}$$

où $h_0(t)$ et $S_0(t)$ correspondent à la fonction de hasard et à la fonction de survie de la distribution baseline (c'est-à-dire, la distribution lorsque tous les coefficients $\boldsymbol{\beta}$ sont égaux à 0).

L'algorithme de Metropolis

L'algorithme de Metropolis est une méthode numérique permettant d'obtenir une séquence de nombres aléatoires provenant d'une distribution de probabilité compliquée. Cette séquence peut être utilisée afin d'approximer la distribution en question. Cet algorithme est souvent utilisé en statistique bayésienne. Voici les étapes à suivre afin d'appliquer cet algorithme (version simplifiée) :

Soit λ le paramètre d'intérêt.

i) Soit λ_t la valeur du paramètre d'intérêt à l'iteration t.

ii) On propose une nouvelle valeur pour le paramètre d'intérêt : $\omega = \lambda_t + z$ avec $z \sim N(0, \sigma_\lambda)$, où σ_λ est l'écart-type de la distribution normale (spécifié au début de l'algorithme).

iii) On génère la probabilité α d'acceptation de cette nouvelle valeur de la manière suivante : $\alpha = \min(1, \exp[\pi(\omega|\mathcal{D}) - \pi(\lambda_t|\mathcal{D})])$, avec $\pi(.|\mathcal{D})$ qui est le logarithme de la distribution postérieure du paramètre d'intérêt.

iv) Soit $U \sim unif(0,1)$, Si $U < \alpha$, alors on pose $\lambda_{t+1} = \omega$, sinon on pose $\lambda_{t+1} = \lambda_t$

v) On retourne à l'étape ii) jusqu'à ce que le nombre d'itérations demandé soit atteint.

Remarque : Voir également le support du cours du 8/11 sur Icampus.
Aide : Pour le point (iii), vous utiliserez le logarithme de la distribution postérieure jointe définie à l'équation 1.

Les données

Dans le fichier *ProjetR.txt* se trouve les données que vous devrez analyser. Ce fichier comporte les informations suivantes :

Nom de la variable	Description	Label
PROV	Province du patient	Néant
T	Temps observé	Néant
CENS	Indicateur de censure (δ)	0 = Censuré, 1 = Observé
AGE	Age du patient (x_1)	Néant
TRT	Traitement reçu par le patient (x_2)	0 = Traitement A ; 1 = Traitement B

Les questions

Partie A : Description des données et préparation des données

1) Créez un répertoire de travail pour le projet R. A l'aide d'une ligne de code, modifiez le répertoire courant de votre session R afin que celui-ci corresponde au répertoire créé préalablement.

2) Importez la base de données du fichier *ProjetR.txt* dans votre session R. Notez qu'il est strictement interdit de modifier manuellement ce fichier.

3) Scindez la base de données en trois bases de données distinctes : une par province (Bruxelles, Brabant Wallon, Flandre).

4) Pour chaque province, effectuez les statistiques descriptives des données. Pour les variables quantitatives continues AGE et T, effectuez uniquement les statistiques descriptives sur les individus non censurés. Présentez les résultats dans un fichier texte portant le nom de la province étudiée. Les résultats à obtenir se trouvent dans les fichiers textes annexés à l'énoncé du projet. Inspirez-vous de la mise en page de ceux-ci afin de présenter clairement les résultats. Faites attention à la gestion des valeurs manquantes.

5) Supprimez les valeurs manquantes des différents jeux de données et standardisez la variable AGE. (Rappel, soit Y une variable. La standardisation de Y est obtenue par la transformation suivante : $\frac{Y-\bar{Y}}{s_Y}$)

Partie B : Illustration graphique des résultats de l'analyse.

Pour les individus provenant de la province de Flandre, nous avons généré un échantillon aléatoire à l'aide de l'algorithme de Metropolis pour chacun des paramètres du modèle (λ, β_1, β_2). Ces échantillons aléatoires se trouvent dans les fichiers *lambda_Flandre.txt*, *beta1_Flandre.txt*, *beta2_Flandre.txt*. Commencez par importer ces trois fichiers dans R.

1) Ecrivez une fonction permettant d'obtenir la Figure 1 :

   – La première ligne reprend la valeur sauvegardée du paramètre versus le numéro de l'itération.
   – La deuxième ligne reprend l'histogramme de l'échantillon postérieur. Sur celui-ci est surperposé une estimation de la fonction de densité postérieure du paramètre.

   La fonction devra sauvegarder le graphique créé dans un document *jpeg* portant le nom *NomDeLaProvince_postérieure.jpeg*.

2) Ecrivez une fonction afin d'obtenir l'estimation de la fonction de survie pour chaque traitement (cf. Figure 2). Celle-ci devra sauvegarder le graphique créé dans un document *jpeg* portant le nom *NomDeLaProvince_survie.jpeg*.
   AIDE : Utilisez la fonction de survie du modèle de Cox en utilisant la médiane de l'échantillon postérieur pour chaque paramètre estimé. Utilisez la mediane de l'âge standardisé comme valeur pour la variable AGE. Evaluer la fonction de survie sur une sequence de valeurs comprises entre 0 et 7.

Partie C : Application de l'algorithme de Metropolis

Dans cette dernière partie, nous vous demandons d'implémenter vous-même l'algorithme de Metropolis pour les individus provenant des provinces de Bruxelles et du Brabant Wallon.

Afin d'effectuer l'analyse, nous allons utiliser la distribution exponentielle de moyenne $\frac{1}{\lambda}$ comme distribution baseline pour le modèle de Cox. Pour rappel, voici les expressions des fonctions de densité, survie et hasard de cette distribution (définies uniquement sur $\mathcal{R}_0^+$) :

$$\mathrm{f}^X(x) = \lambda \exp(-\lambda x) \qquad \mathrm{S}^X(x) = \exp(-\lambda x) \qquad \mathrm{h}^X(x) = \lambda$$

1) Voici le logarithme de la distribution postérieure à utiliser dans l'algorithme de Metropolis :

$$\pi(t, \lambda, \beta_1, \beta_2 | \mathcal{D}) \propto \sum_{i=1}^{n} \{\delta_i \log[h(t_i)] + \log[S(t_i)]\} - \log[\lambda] \tag{1}$$

   Remarque : Les fonctions de hasard et survie définies dans le logarithme de la distribution postérieure sont celles du modèle de Cox.

Ecrivez une fonction retournant la valeur du logarithme de la distribution postérieure à un point $(\lambda, \beta_1, \beta_2)$ donné. Cette fonction prendra les paramètres suivants :

- Le vecteur des temps observés T
- Le vecteur des indicateurs de censure δ
- Une valeur du paramètre λ
- Une valeur du vecteur (β_1, β_2)
- Une matrice X dont la première colonne correspond à l'âge (standardisé) du patient et dont la deuxième colonne correspond au traitement reçu.

2) Ecrivez une fonction prenant comme paramètres :

- Le nombre d'itérations (M = 10000)
- Le nombre de paramètres
- Le vecteur des temps observés T
- Le vecteur des indicateurs de censure δ
- La valeur initiale du paramètre λ qui est égale à 1
- La valeur initiale de (β_1, β_2) qui vaut $(0.4, -0.4)$
- La matrice X utilisée à la question précédante.
- Un vecteur reprenant l'écart-type de la distribution normale utilisée dans l'algorithme de Metropolis pour chaque paramètre. ($\sigma_\lambda = 0.2$, $\sigma_{\beta_1} = 0.19$, $\sigma_{\beta_2} = 0.27$)

et définissant l'algorithme de Metropolis. Cette fonction devra retourner les informations suivantes :

- L'échantillon aléatoire postérieur du paramètre λ
- L'échantillon aléatoire postérieur du paramètre β_1
- L'échantillon aléatoire postérieur du paramètre β_2
- Le taux d'acceptation de chaque paramètre.

AIDE : Voici comment structurer votre fonction :

- Appliquez les 4 premières étapes de l'algorithme de Metropolis au paramètre λ.
- En utilisant la valeur sauvegardée de λ, appliquez les 4 premières étapes de l'algorithme de Metropolis au paramètre β_1
- En utilisant la valeur sauvegardée de λ et β_1, appliquez les 4 premières étapes de l'algorihme de Metropolis au paramètre β_2
- Englobez le tout dans une boucle afin de réaliser le nombre d'itérations requis.

3) En utilisant les résultats obtenus, appeler les deux fonctions de la partie B afin d'obtenir l'illustration graphique de votre analyse.

Répartition des points

ATTENTION : Si une ou plusieurs erreurs (de syntaxe, variables non déclarées...) sont détectée(s) dans le code lors de l'exécution de celui-ci, la note du projet sera de 0/3 ! ! ! La seule modification que le correcteur appliquera à votre code sera la modification du répertoire de travail.

	Partie A (8 POINTS)	**Partie B (6 POINTS)**	**Partie C (6 POINTS)**
1)	1 POINT	3 POINTS	2 POINTS
2)	1 POINT	3 POINTS	3 POINTS
3)	1 POINT		1 POINT
4)	3 POINTS		
5)	2 POINTS		

L'efficacité, la lisibilité et les commentaires de votre programme seront pris en compte lors de la correction. Evitez les copier-coller inutiles.

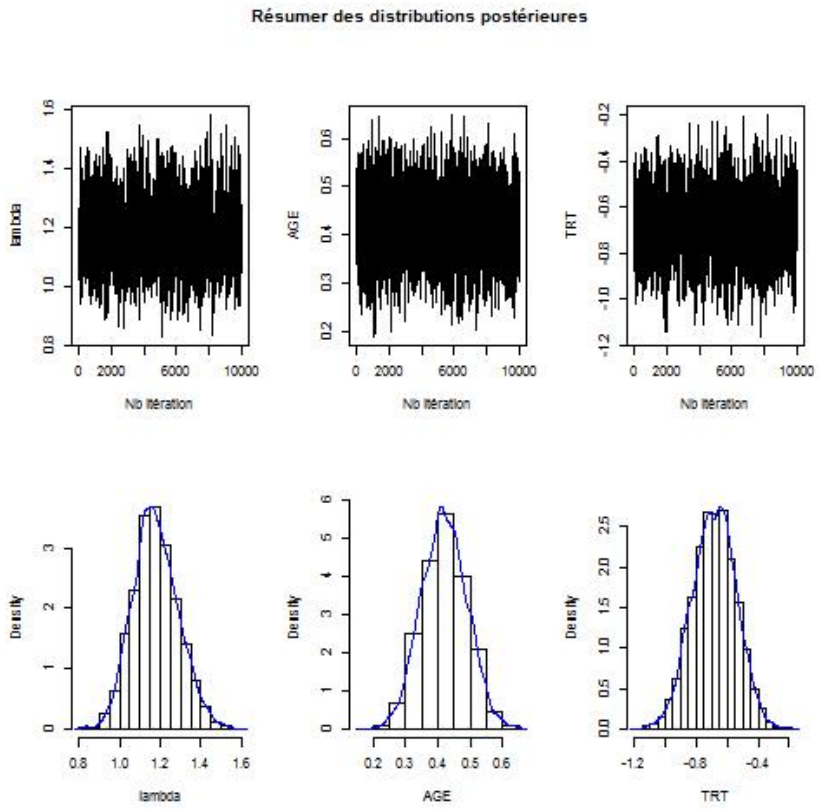

FIGURE 1 – Résumer des distributions postérieures.

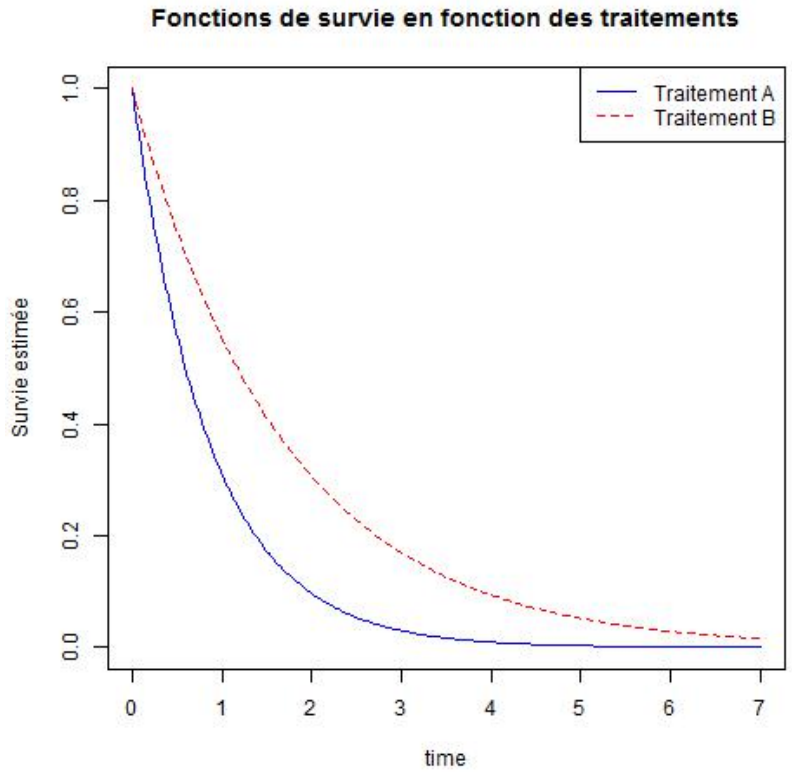

FIGURE 2 – Estimation des fonctions de survie.