**ENSIMAG 2ème année** **7-11 février 2000**
**Option Mathématiques - Informatique**

**Stage de programmation scientifique**
**TP - Introduction à Splus**

**Présentation** Le logiciel S (S-Plus) propose un environnement puissant pour traiter ou manipuler des données. Il est constitué d'une suite intégrée de moyens destinés à l'analyse des données et à leur visualisation. Il contient une collection d'outils pour les statistiques et l'ADM, un langage pour l'implantation de modèles statistiques, un environnement graphique et un langage de programmation "orienté objet". La plupart des entités créées en S sont permanentes. Ces entités sont les objets : données, résultats, fonctions. Elles sont stockées dans le répertoire .Data. Ceci signifie que le résultat d'une procédure statistique peut être réutilisé lors de différentes sessions. Pour cette raison, il est important de créer un répertoire pour chaque projet statistique effectué en S (mkdir TpStat) ainsi qu'un sous-répertoire TpStat/.Data que S utilisera par défaut. Une session de S-Plus s'ouvre par la commande

```
peggy$ SPlus
```

(ou S si S-Plus n'est pas disponible) et se termine par

```
> q()
```

S-Plus possède une documentation en ligne que l'on ouvre avec

```
> help.start(gui="motif")
```

L'environnement graphique de Splus est initialisé grâce à

```
> motif()
```

Techniquement, S est un langage fonctionnel. Les commandes élémentaires sont constituées d'expressions et d'affectations.

```
> 2 + 2
[1] 4
> a <- c(1,4,2,3)
> a
[1] 1 4 2 3
> mean(a)
[1] 2.5
> a + 1
[1] 2 5 3 4
> a[2:4]
[1] 4 2 3
> a[ a < 3 ]
[1] 1 2
```

S peut être complété en écrivant de nouvelles fonctions. Voici un exemple. On souhaite calculer la statistique

$$\text{stat.log}(x) = -\frac{1}{n}\sum_{i=1}^{n}\log(x_i)$$

où $0 < x_i < 1$ pour tout $i \leq n$.

```
> stat.log <- function(x) {
+ n <- length(x)
+ it <- 0
+ stat.log <- 0
+ while ( (it <- it+1) <= n) {
+ stat.log <- stat.log + log(x[it])
+ }
+ - stat.log/n
 }
```

La fonction stat.log pourra être désormais utilisée comme une fonction standard de S. Pour editer en S, on utilisera vi. Par exemple, admettons que l'on ait créé la fonction

```
> f <- function(x) { x + 1 }
```

la commande (vi(f) $- >$ f) permet de lancer une session de vi dont on sortira en sauvegardant les modifications effectuées pour f.

Le travail demandé ci-dessous est constitué de deux exercices indépendants portant sur la simulation et l'estimation. Le premier exercice porte sur les vecteurs gaussiens et la régression linéaire. Le second sur l'estimation par maximum de vraisemblance. Le compte rendu du Tp devra inclure listing, sorties et commentaires.

## Exercice I

Il s'agit dans ce premier exercice de simuler 100 réalisations de la loi gaussienne à deux dimensions de moyenne nulle et de matrice de covariance $K = A\,{}^{T}\!A$ où

$$A = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$$

On visualisera ces réalisations puis on tracera la droite de régression théorique

$$y = \frac{\text{Cov}(Y_1, Y_2)}{\text{Var}(Y_1)}x$$

où $(Y_1, Y_2)$ est une vecteur gaussien de moyenne et de covariance souhaitées, ainsi que la droite estimée par la méthode des moindres carrés. L'exercice est prétexte à manipuler des matrices.

1. Créer un vecteur de données $x$ comportant 200 réalisations de la loi normale centrée réduite (rnorm). Créer la matrice $A$ (matrix).

2. Créer l'échantillon $X$ (à partir de $x$) de loi gaussienne $\mathcal{N}(0, I)$ et l'échantillon $Y$ de loi $\mathcal{N}(0, K)$. (Rappel : Y peut être obtenu par la transformation linéaire $Y = AX$.) Le produit matriciel en S s'écrit

```
A %*% B
```

3. Visualiser l'échantillon $Y$ (x11, plot).
4. Définir une fonction reg.theo de $A$ qui calcule le coefficient de régression théorique et visualise la droite de régression (utiliser abline et t(A) pour calculer la transposée de $A$).
5. Tracer la droite de régression empirique (moindres carrés) à l'aide de lm et abline.
6. Visualiser l'histogramme des données (utiliser hist2d).

## Exercice II

L'objectif de cet exercice et de simuler un échantillon de la loi de Poisson "tronquée" (c'est à dire la loi conditionnelle sachant que la variable est non nulle) puis d'estimer le paramètre de cette loi par la méthode de maximum de vraisemblance. Soit $\lambda > 0$. L'expression de la loi est

$$\forall x \geq 1 \ , \quad \mathrm{P}(X = x) = \frac{e^{-\lambda}}{1 - e^{-\lambda}} \frac{\lambda^x}{x!}$$

L'estimateur de maximum de vraisemblance $\hat{\lambda}$ vérifie (le montrer)

$$\bar{x} = \frac{\hat{\lambda}}{1 - e^{-\hat{\lambda}}}$$

Ecrire une procédure qui résout numériquement cette équation (méthode de Newton) et qui affiche le résultat pas par pas (cat). Tester cette procédure sur les données simulées. L'estimateur est-il biaisé ?