G. Hunault
Licence Informatique

Angers, octobre 2003

Contrôle continu pour l'Option de Probabilités et Statistiques

Modalités du Contrôle

Le contrôle continu comprend deux petits projets à rendre, l'un concernant la partie Probabilités, l'autre la partie Statistiques. Chaque étudiant(e) doit rendre les deux projets. Chaque projet comporte une partie programmation ET une partie rédaction **qui ne comptent pas à part égale** dans la note car la rédaction est considérée comme plus importante que le programme.

La rédaction se fait au choix mais exclusivement sous *Word* ou sous *Latex*. La programmation est effectuée au choix mais exclusivement en *Awk*, *Apl*, *Rexx*, *C*, *C++*, *Perl*, *Tcl*, *Java* ou *Pascal*. En plus des algorithmes et programmes, on pourra ne rendre qu'un seul document qui contiendra les explications sur les programmes et les réponses aux questions pour les deux projets. Tout le travail (explications, algorithmes et programmes) sera fourni sous forme de fichiers informatiques et transmis par *mail* sous forme d'une archive informatique. Cette archive, unique, sera au format .zip ou .tar et les fichiers y seront archivés sans sous-répertoire. Ce fichier devra être remis au plus tard le

lundi 08 décembre, à 14 h dernier délai

Pour éviter toute ambiguïté, tous les noms de fichiers et en particulier celui de l'archive, commenceront par les initiales de l'élève (document signé en TD). Il sera possible de vérifier auprès de l'enseignant le lundi 15 décembre entre 14 et 18 h que l'archive a bien été reçue et que les fichiers ont bien été décompressés et qu'ils sont lisibles, compilables etc.

1. Projet de Probabilités

Une société qui vend des CDROM veut faire une offre promotionnelle avec un cadeau en fonction d'un code écrit sur l'étiquette à coté du code-barres et du prix du CD. Ne maîtrisant pas bien le calcul des probabilités, cette société décide d'écrire un programme pour gérer les codes et compter les codes gagnants. Voici les règles de définition des codes :

- un code contient de 1 à 17 lettres majuscules sans accent prises parmi les 26 lettres de l'alphabet français standard ;
- un code ne contient que des lettres.

A chaque code correspond un score, défini comme suit

- le score d'un code correspond à la somme des scores des lettres du code augmenté d'un bonus pour la première lettre et d'un bonus pour la dernière lettre ;
- le bonus vaut 2 pour une consonne en première lettre, 0 pour une voyelle en première lettre ;
- les voyelles AEIOU ont comme score 5, la voyelle Y a comme score 3 ;
- les consonnes JKQWXZ ont comme score 1, les autres consonnes ont comme score 2 ;
- le bonus vaut 3 si la dernière lettre est S, 1 si c'est une voyelle ou X, 0 sinon.

Voici quelques exemples de score

```
Code                Score correspondant

S                      7
ZSU                   11
AVION                 19
BATEAUX               28
CONFITURE             33
```

Le travail demandé consiste à écrire un algorithme qui en fonction du nombre de lettres passé en paramètre donnera la probabilité du plus petit nombre de mots en longueur fixe avec ce nombre de lettres et ayant le même score. On affichera bien sûr ce score ainsi que le nombre de chaînes correspondant et un exemple de code pour ce score. Par exemple si l'utilisateur a passé en paramètre la valeur 2 alors le programme correspondant à l'algorithme affichera

```
     nbLet nbMot   score       proba     exemple
         2    21       8       0.0311    BY
```

ce qui signifie que le plus petit nombre de mots de 2 lettres exactement ayant le même score est 21, que ce score est 8, que sa probabilité à 10^{-4} près est 0.0311 et qu'il est obtenu notamment pour le code BY.

Il vous faudra ensuite traduire l'algorithme dans un langage de votre choix parmi les langages cités puis rédiger la documentation de votre algorithme et de votre programme. Enfin, pour conclure ce projet de probabilités, vous essaierez de répondre aux questions suivantes en justifiant vos réponses et vos sources (ce pourra être des URL, ce qui vous dispense de faire du copier/coller) :

- qu'affiche le programme pour les mots d'une seule lettre ? et pour les mots de 8 lettres exactement ?
- y a-t-il une formule mathématique pour trouver le plus petit nombre de mots ayant le même score en fonction du nombre de lettres ? si oui, laquelle ? si non, pourquoi ?
- est-ce que les entreprises qui font des cadeaux pour certains codes utilisent la même approche et pourquoi ?
- combien de temps met votre programme pour les mots de 12 lettres ?

L'affichage présenté est un modèle à respecter scrupuleusement ce qui permet (via la redirection de la sortie standard avec >) d'obtenir à peu de frais un tableau des différentes valeurs en fonction du nombre de lettres, tableau bien cadré unité sous unité dizaine sous dizaine etc. Pour vérification :

```
␣␣␣␣␣nbLet␣nbMot␣␣␣score␣␣␣␣␣␣␣proba␣␣␣␣␣exemple
␣␣␣␣␣␣␣␣␣2␣␣␣␣21␣␣␣␣␣␣␣8␣␣␣␣␣␣␣0.0311␣␣␣␣BY
```

2. Projet de Statistiques

Lorsqu'une grandeur comme le poids d'un individu, le rendement d'une parcelle de blé varie en fonction d'un seul paramètre qualitatif, par exemple la ville où réside l'individu ou le type d'engrais utilisé sur la parcelle, il est d'usage en statistique d'effectuer un type de régression linéaire particulier nommé analyse de la variance à un facteur. Les données d'entrée sont en général présentées comme pour le fichier suivant nommé `ble.txt`

```
1   1   45
1   2   31
1   3   32
1   4   39
2   1   42
2   2   34
2   3   33
2   4   38
3   1   39
3   2   40
3   3   38
3   4   41
4   1   41
4   3   33
4   4   44
5   1   48
```

Chaque ligne du fichier doit être interprété comme suit : le premier mot indique le numéro de la mesure, le deuxième mot indique le numéro de facteur et enfin le troisième mot est la valeur de la mesure.

A ce fichier de données correspond un tableau de mesures par facteur qui s'en déduit facilement quand on sait que les facteurs sont mis en colonne, à savoir

```
45    31    32    39
42    34    33    38
39    40    38    41
41          33    44
48
```

On note alors pour ce qui suit $x_{i,j}$ la valeur en ligne i colonne j de ce tableau de mesures par facteur.

On définit et on calcule ensuite les quantités suivantes :

- n_j est l'effectif de la colonne j c'est à dire le nombre de valeurs non nulles de cette colonne ;

- m_j est la moyenne arithmétique de la colonne j c'est à dire le rapport "somme des valeurs non nulles sur le nombre de valeurs non nulles" pour cette colonne ;

- σ_j est l'écart-type de la colonne j calculé selon la formule "racine carrée de la variance" où la variance est "la moyenne des carrés moins le carré de la moyenne" ;

- c_j est le coefficient de variation de la colonne j soit le rapport "écart-type sur moyenne" exprimé en pourcent ;

- N est l'effectif total c'est à dire le nombre de valeurs non nulles du tableau ;

- n est le nombre de colonnes non nulles présentes dans le tableau ;

- m est la moyenne arithmétique général de l'ensemble des valeurs non nulles du tableau ;

- σ est l'écart-type général de l'ensemble des valeurs non nulles du tableau ;

- c est le coefficient de variation de l'ensemble des valeurs non nulles du tableau ;

- SCEF est la somme (pondérée) des carrés des écarts factoriels définie par la formule
$$\sum_j n_j.(m_j - m)^2$$

- SCER est la somme des carrés des écarts résiduels définie par la formule
$$\sum_{i,j} (m_j - x_{i,j})^2$$

- STCE est la somme totale des carrés des écarts définie par la formule
$$\sum_{i,j} (m - x_{i,j})^2$$

Enfin, on associe aux sommes SCEF, SCER et STCE des degrés de liberté à savoir respectivement $n-1$, $N-n$ et $N-1$ et on calcule les variances empiriques correspondantes à savoir

$$\begin{aligned} V_f &= \text{SCEF} \;/\; n-1 \\ V_r &= \text{SCER} \;/\; N-n \\ V_t &= \text{STCE} \;/\; n-1 \end{aligned}$$

ce qui permet (ouf !) de calculer le rapport de variance R_v défini par

$$R_v = STCE/SCER$$

Pour réaliser le projet de statistiques vous viendrez dans un premier temps écrire un algorithme qui lit un fichier-texte dont le nom est passé en paramètre, sachant que ce fichier contient un nombre variable de lignes avec 3 mots comme pour le fichier `ble.txt` ; on utilisera un deuxième paramètre k qui servira pour les affichages (voir ci-dessous).

L'algorithme doit ensuite réafficher le tableau de mesures de façon plus lisible puis calculer les quantités définies, sachant que les valeurs et les résultats statistiques (sauf les effectifs et les résultats généraux) utilisent k décimales.

Vous viendrez ensuite traduire l'algorithme dans un langage de votre choix parmi les langages cités puis rédiger la documentation de votre algorithme et de votre programme. Vous rédigerez ensuite une page entière sur le ou les inventeurs de la méthode et de ce type de présentation, les intérêts (cachés) de cet affichage, la disponibilité de cet affichage dans les logiciels statistiques usuels sans oublier de citer précisément vos sources. Celles-ci pourront être des URL, ce qui vous dispense de faire du copier/coller.

Vous fournirez également l'affichage obtenu si on exécute votre programme en lui passant comme paramètres `betail.ano 3`. Le fichier de données correspondant est disponible à l'adresse

http://www.info.univ-angers.fr/pub/gh/wstat/betail.ano

ainsi que sur le disque K: des machines de la Faculté des Sciences, dans le sous-répertoire Stat_Ad.

On trouvera aussi à ces adresses le fichier `ble.txt`.

Vous terminerez ce travail de statistiques en répondant aux quatre questions

- pourquoi utilise-t-on le terme "anova" en anglais pour désigner l'analyse de la variance ?
- à quelle partie du cours (ou plutôt des TD) ce genre de calcul fait-il référence ? à quelle loi théorique célèbre et avec quels degrés de liberté faudrait-il comparer R_v ?
- après avoir rapidement utilisé deux logiciels statistiques que vous aurez soigneusement choisi et que vous citerez explicitement, quelle valeur pour R_v trouve-t-on ? s'il y a une différence avec votre valeur, vous essaierez de l'expliquer puis vous indiquerez quel logiciel vous préférez en justifiant votre choix.
- qu'apporterait le test de Newman-Keuls si on l'appliquait à nos données ?

Afin de pouvoir vérifier les calculs, on trouvera sur la page

`http://www.info.univ-angers.fr/pub/gh/wstat/resanova.htm`

les résultats d'exécutions souhaités, bien cadrés, décimale sous décimale, unité sous unité etc. pour le fichier `ble.txt` avec comme deuxième paramètre la valeur 1 puis 2 puis 3. Cet affichage constitue un modèle à respecter scrupuleusement.

On pourra admettre que toutes les données sont positives strictement et inférieures à 2003, qu'il y a au plus 15 facteurs et au plus 25 mesures par facteur.

Remarques concernant la rédaction

La note sera principalement fonction de la qualité de l'analyse du travail effectué, de l'exposé de la méthode et de la clarté de la rédaction. La valeur attribuée au programme ne représentera que 25 % de la note. Aucune interface n'est demandée, les programmes devront être exécutables en ligne de commande (à cause des paramètres). En cas de langage compilé, on fournira impérativement le programme-source et l'exécutable produit. On précisera, si besoin est, la commande de compilation. En cas de langage interprété, on fournira impérativement le programme-source. Dans tous les cas de figures, on précisera la version du langage choisi, le type d'ordinateur et le système d'exploitation utilisés.

Aucun langage autre que cité ne devra être utilisé, aucun traitement de textes autre que cité ne devra être utilisé. Tout fichier dont le nom ne commencera pas par les initiales de l'élève comme signé en TD ne sera pas lu. En particulier, les fichiers nommés uniquement proba, algo, stat,projet, doc ne seront pas lus.

Un programme même incomplet mais commenté et bien indenté sera au moins autant apprécié qu'un programme qui "marche" mais sans commentaire et illisible quant aux structures de flot (boucles, tests etc.). Il est conseillé d'écrire un programme court avec des sous-programmes bien paramétrés, notamment pour l'affichage soigné avec k décimales.

Les identificateurs auront tous au moins 4 caractères. Les algorithmes utiliseront la syntaxe de *Galg*. On fournira d'ailleurs non pas le texte original des algorithmes mais leurs résultats d'analyse par *Galg* avec le numéro de ligne, le numéro d'instruction, l'indication de profondeur etc. et le fichier des tableaux, variables et modules. Compte-tenu de l'absence véritable de difficultés dans les algorithmes à programmer, les pointeurs sont totalement déconseillés et même interdits sauf à inventer et rédiger les instructions algorithmiques correspondantes et leur syntaxe, instructions à justifier et à incorporer dans *Galg*.

Il est très fortement recommandé d'utiliser pour la toute dernière version du document rédigé un correcteur orthographique et grammatical, chaque faute d'orthographe (ou de frappe) détectable par ordinateur étant pénalisée.

3. Références

Ce texte, de même que le cours de *Probabilités et Statistiques* pour la licence informatique, les T.D. correspondants (avec solution) et la procédure pour envoyer le mini-projet peut être récupéré à partir de l'adresse

`http://www.info.univ-angers.fr/pub/gh/wstat/`

Un manuel élémentaire d'*Algorithmiques raisonnées* et les références à *Galg* sont disponibles depuis la page

`http://www.info.univ-angers.fr/pub/gh/wals/`

De plus, il est possible de tester si le projet a été correctement reçu grâce à la page

`http://www.info.univ-angers.fr/pub/gh/wstat/zipdirpli.php3`