Traitement des données numérique

Travail 2 - Partie chef de projet

Détecter les anomalies

Année académique 2017 - 2018

Informatique de gestion, 2ème Groupe A

A l'attention de Madame Charlier

Samuel **Decruynaere**

Table des matières

Enoncé

Le but de ce travail est de réaliser un programme comportant les fonctionnalités suivantes :

- décrypter la clé secrète de Shamir ;
- calculer les coordonnées du point max (global) d'un polynôme d'interpolation ;
- calculer a^x sur son domaine et l'afficher;
- détecter les anomalies (1 variable) : construire le modèle ;
- détecter les anomalies (1 variable) : utiliser le modèle.

L'utilisateur devra choisir dans un menu reprenant ces différentes propositions (ce code vous est fourni).

NB :

- les noms écrits en gras sont des noms de variables à respecter ;
- réutiliser les librairies implémentées dans le chapitre 1.

Détecter les anomalies (1 variable)

In : fichier **fiModele.xlsx** (contenant les observations selon une variable numérique) permettant d'établir le modèle ;
le niveau d'incertitude pour les limites de contrôle (**alphaControl**) et le niveau d'incertitude pour les limites de surveillance (**alphaWarning**) ;
n le nombre de valeurs que contiendront les échantillons dans la phase d'exploitation.

Out : Les paramètres de la loi normale qui sert de modèle : **moyenne**, **ecartType**, les limites de contrôle (**lower(upper)ControlLimit**) et les limites de surveillance (**lower(upper)WarningLimit**).

Pour ce faire, vous pouvez utiliser la librairie « simpson.h » se trouvant sur le commonProfsEtudiants/Charlier.c/biblioInteg.
Voici le contenu de « simpson.h » :

typedef double(*Fonction)(double);
double calculSimpson(int , double, double, Fonction);

paramètres d'entrée : le nombre de points (nbPoints), la borne inférieure (borneInf), la borne supérieure (borneSup) ainsi qu'un pointeur vers la fonction dont on veut calculer l'intégrale ;

paramètre de sortie : la valeur de l'intégrale calculée par la méthode de Simpson

ATTENTION : chaque fonction devra faire l'objet d'un testing complet (cfr p4).

Diagramme d'action

```
entrée : fiModele.xlsx
┌── *
│ obtenir n = 0
│ obtenir alphaControl // 5%
│ obtenir alphaWarning // 1%
│ sommeXi = 0
│ sommeXiCarré = 0
│ o──────────────────────────────────────o ↓  *f, nbPts
│ │ construction de la table ( Simpson ) │
│ o──────────────────────────────────────o ↓  tabNormale
│ xi = lire dataFile
│ ╔══ do while (!EOF)
│ ╠ sommeXi += xi
│ ╠ sommeXiCarré += xi * xi
│ ╠ xi = lire dataFile
│ ╠ n++
│ ╚──
│ moyenne = somme xi / n
│ variance = sommeXiCarré / n - (moyenne * moyenne)
│ o──────────────────────────────────────────o ↓  alphaControl, tabNormale
│ │ recherche de la valeur en fonction de alpha │
│ o──────────────────────────────────────────o ↓  coéfficientAlphaControl
│
│ o──────────────────────────────────────────o ↓  alphaWarning, tabNormale
│ │ recherche de la valeur en fonction de alpha │
│ o──────────────────────────────────────────o ↓  coéfficientAlphaWarning
│
│ o──────────────────────o ↓  moyenne, variance, n, coéfficientAlphaControl
│ │ calcul de l'intervalle │
│ o──────────────────────o ↓  lowerControlLimit, upperControlLimit
│ sortie de l'intervalleControle
│
│ o──────────────────────o ↓  moyenne, variance, n, coéfficientAlphaWarning
│ │ calcul de l'intervalle │
│ o──────────────────────o ↓  lowerWarningLimit, upperWarningLimit
│ sortie de l'intervalleSurveillance
│
│ o──────────────────────────────o ↓ n,lowerControlLimit,upperControlLimit,lowerWarningLimit,upperWarningLimit,dataAEvaluer
│ │ traitement sur base du modèle │
│ o──────────────────────────────o
└──
```

```
┌── * construction de la table ( Simpson )
│ // comme la loi normale est symétrique j'ai choisi de construire mon tableau a partir de x = 0;
│ nbPts = 3 // a vérifier si précision suffisante
│ dernierRésultat = 0.5
│ tabNormal[0][0] = dernierRésultat
│ bInf = 0
│ iLigne = 0
│ ╔══ do while (iLigne < 40) // attention travailler en entier
│ ╠ iColonne = 0
│ ╠ ╔══ do while (iColonne < 10) // attention travailler en entier
│ ╠ ╠ bSup = iLigne / 10 + iColonne / 100
│ ╠ ╠ o─────────o ↓  nbPts, bInf, bSup, *f // f est un pointeur vers l'inégrale de gauss
│ ╠ ╠ │ Simpson │
│ ╠ ╠ o─────────o ↓  resultat
│ ╠ ╠
│ ╠ ╠ dernierRésultat += resultat
│ ╠ ╠ tabNormal[iLigne][iColonne] = resultat * 10000 // pour ne pas avoir de problème de conversion binaire mais ???
│ ╠ ╠ bInf = bSup;
│ ╠ ╠ iColonne++
│ ╠ ╚──
│ ╠ iLigne++
│ ╚──
└──
```

```
┌── * recherche de la valeur en fonction de alpha
│ interpolation = 0
│ zoneNormaliser = (1 - alpha / 200) * 10000
│
│ iLigne = 0
│ ╔══ do while (iLigne < 39 AND tabNormale[iLigne + 1][0] ≤ zoneNormaliser)
│ ╠ iLigne++
│ ╚──
│
│ iColonne = 0
│ ╔══ do while (iColonne < 9 AND tabNormale[iLigne][iColonne + 1] ≤ zoneNormaliser)
│ ╠ iColonne++
│ ╚──
│
│ ┌── if (tabNormale[iLigne][iColonne] ≠ zoneNormaliser)
│ ├ iLigneBord = iLigne
│ ├ iColonneBord = iColonne + 1
│ ├── if (iColonne == 9 AND tabNormale[iLigne][iColonne] <  zoneNormaliser)
│ ├ iLigneBord = iLigne + 1
│ ├ iColonneBord = 0
│ └──
│ interpolation = (zoneNormaliser - tabNormale[iLigne][iColonne])
│ interpolation /= (tabNormale[iLigneBord][iColonneBord] - tabNormale[iLigne][iColonne])
└──
coéfficientAlpha = iLigne / 10 + iColonne / 100 + interpolation / 100
```

```
* calcul de l'intervalle
margeErreur = coefficentAlpha * sqrt(variance / n);
intervalleInf = moyenne - margeErreur;
intervalleSup = moyenne + margeErreur;
```

```
* traitement sur base du modèle
sommeEchantillon = 0
xi = lire dataAEvaluer
do while (!EOF)
    do while (n Times)
        sommeEchantillon += xi
        xi = lire dataAEvaluer

    xBarre = sommeEchantillon / n

    if (xBarre < lowerWarningLimit || xBarre > upperWarningLimit)
        texte = "nous sommes en dehors de l'intervalle de surveillance"
        if (xBarre < lowerControlLimit || xBarre > upperControlLimit)
            texte = "valeur érroné"

        sortir texte
```

Planning

Construction de la table et Recherche de la valeur en fonction de alpha

Pour le 23 avril par Romain Demoustiez.

Calcul de l'intervalle et traitement sur base du modèle

Pour le 23 avril par Alexandre Eeckhout.

Détecter les anomalies

Pour le 25 avril par Alexandre Eeckhout.

Le travail final doit être rendu le 27/4.

Testing

échantillon de base

Volume (xi)	ri
20 m³	8
40 m³	16
60 m³	12
80 m³	2

Moyenne : 44.21053 - **Variance** : 277.00831

AlphaControl : 1% - **AlphaWarning** : 5%

n = 38

Recherche de la valeur dans la table normal en fonction de alpha

Libellé du test	Argument encodé	Nb décimales	Résultat attendu	Résultat obtenu	Conclusion (ok ?)
1%	(1, tabNormal)	5	2.57500		
4.6%	(4.6, tabNormal)	5	1.99600		
100%	(100, tabNormal)	5	0.00000		

Calcul de l'intervalle

Libellé du test	Argument encodé	Nb décimales	Résultat attendu	Résultat obtenu	Conclusion (ok ?)
10%	(Moyenne, Variance, n, 1.64500)	5	[39.77, 48.65]		

Traitement sur base du modèle

n, lowerControlLimit, upperControlLimit, lowerWarningLimit, upperWarningLimit, dataAEvaluer

lowerControlLimit (1%) : 37.26 - **upperControlLimit (1%)** : 51.16

lowerWarningLimit (5%) : 38.92 - **upperWarningLimit (5%)** : 49.50

n : 38

Libellé du test	Argument encodé	Nb décimales	Résultat attendu	Résultat obtenu	Conclusion (ok ?)
Lower Warning Limit	(38, 37.26, 51.16, 38.92, 49.50, **38.92**)	5			
	(38, 37.26, 51.16, 38.92, 49.50, **39.00**)	5	"nous sommes en dehors de l'intervalle de surveillance"		
Upper Warning Limit	(38, 37.26, 51.16, 38.92, 49.50, **49.50**)	5			
	(38, 37.26, 51.16, 38.92, 49.50, **50.00**)	5	"nous sommes en dehors de l'intervalle de surveillance"		
Lower Control Limit	(38, 37.26, 51.16, 38.92, 49.50, **37.26**)	5	"nous sommes en dehors de l'intervalle de surveillance"		
	(38, 37.26, 51.16, 38.92, 49.50, **37.00**)	5	“valeur érroné”		
Upper Control Limit	(38, 37.26, 51.16, 38.92, 49.50, **51.16**)	5	"nous sommes en dehors de l'intervalle de surveillance"		
	(38, 37.26, 51.16, 38.92, 49.50, **52.00**)	5	“valeur érroné”		
Moyenne	(38, 37.26, 51.16, 38.92, 49.50, 44.21053)	5			