TD Manipulations de données sous SAS – Partie 1

Aucune proc SQL ne doit être utilisée lors de ce TD.

Exercice 1

1. A l'aide de la commande **DATA**, créer une table SAS nommée « notes » avec le bon format des variables et les données ci-dessous en utilisant un espace comme délimiteur : *(Diapo 9)*

	nom	semestre	a_naiss	moyenne
1	Ricco	S1	1953	7
2	Julien	S2	1967	12
3	Omar	S1	1940	3
4	Fadila	S1	1973	14

2. Faire un nouveau **DATA** nommé « notes_2 ». Nous garderons le même code que celui de la question 1 en changeant le délimiteur en virgule. *(diapo 13)*
3. Pour cette question, nous allons faire un **DATA** nommé « notes_2 » (il écrasera celui fait précédemment).
   Augmenter la taille de la variable nom pour insérer le nom de famille des étudiants inscrits. (**LENGTH**) L'espace n'est plus considéré comme délimiteur :

	nom
1	Ricco Rakotomalala
2	Julien Jacques
3	Omar Boussaid
4	Fadila Bentayeb

4. A présent, modifier la date de naissance (a_naiss) des données de **DATA** afin d'y ajouter le jour et le mois : (**DATA**, **SET**)

Attention ! la date s'affiche en calculant le nombre de jours après le 1er Janvier 1960 (Positif) ou avant le 1er Janvier 1960 (Négatif). Utiliser le format : a_naiss DDMMYY10. (**FORMAT**,**INFORMAT**)

a_naiss
23/10/1953
12/03/1967
29/06/1940
24/09/1980

Indication : les données dates peuvent s'écrire de différentes manières :

```
Ricco Rakotomalala,S1,23.10.1953,7
Julien Jacques,S2,12031967 ,2
Omar Boussaid,S1,29/06/1940,3
Fadila Bentayeb,S1,24/09/1980,14
```

5. Insérer un nouvel étudiant SANS utiliser de proc sql : utiliser **IF**, **END=EOF** et la fonction mdy(). (diapo 15)

6. Insérer une colonne nommée bonus, en insérant les bonus suivants : (**DATA**, **SET**)

	nom	semestre	a_naiss	moyenne	bonus
1	Ricco Rakotomalala	S1	23/10/1953	7	2
2	Julien Jacques	S2	12/03/1967	2	3
3	Omar Boussaid	S1	29/06/1940	3	-1
4	Fadila Bentayeb	S1	24/09/1980	14	2
5	Julien Lemaire	S2	06/01/2000	15	2

Puis, calculer une nouvelle colonne : moyenne_finale = moyenne+bonus.

7. Supprimer la colonne semestre SANS PROC SQL ! (**DATA** … (**DROP**=…)) *(Diapo 14)*

Supprimer l'étudiant de la ligne 2 avec l'index. (**IF, _N_=2 THEN… ,** *diapo 15*). (A ce stade, nous avons 4 lignes et 5 colonnes dans notre jeu de données).

8. Supprimer la ligne en sélectionnant le prénom 'Omar Boussaid'. (**IF, THEN**)
9. Renommer la colonne a_naiss en date_naiss.

Faire une nouvelle colonne « avis » en fonction de la moyenne_finale. Si moyenne_finale<10 : ne passe pas, sinon : passe.

Résultat final :

	nom	date_naiss	moyenne	bonus	moyenne_finale	avis
1	Ricco Rakotomalala	23/10/1953	7	2	9	ne passe pas
2	Fadila Bentayeb	24/09/1980	14	2	16	passe
3	Julien Lemaire	06/01/2000	15	2	17	passe

10. Faire un nouveau dataset nommé « note » avec la moyenne, la moyenne_finale et l'avis. (**DATA**, **SET**, **KEEP**)

	moyenne	moyenne_finale	avis
1	7	9	ne passe pas
2	14	16	passe
3	15	17	passe

Exercice 2

1. Créer une bibliothèque dans laquelle vous enregistrerez vos sorties. Rafraîchir l'explorateur de dossier, et veiller à ce que la bibliothèque s'affiche bien. (Possible de le faire depuis l'explorateur, catégorie "Mes Bibliothèques", icône "Nouvelle Bibliothèque" ou voir dans le cours via la fonction LIBNAME, attention au nom du chemin !)

Résultat :

2. Importer le fichier "sise_2019.txt". (PROC IMPORT)

3. Afficher les informations de la table. (PROC CONTENTS)

4. Renommer la variable CP en Code_Postal. (Option RENAME dans une étape DATA)

5. Créer une variable "Appellation" correspondant à la première lettre du prénom suivi d'un point et espacé du nom de famille. Exemple : Jean Bonneau devient J. Bonneau. (SUBSTR pour l'extraction de chaîne, || pour la concaténation)

6. Créer une variable "BAC" qui suit une loi normale de moyenne 15 et d'écart-type 1,8. Veiller à ce que les valeurs ne dépassent jamais 20. (Fonction RAND avec comme argument GAUSSIAN, voir du côté de IF pour ne pas dépasser la note de 20)

7. Formater le nombre de chiffres après la virgule de la variable "BAC" à 2. (Option FORMAT dans une étape DATA)

8. Calculer l'âge de chacun et stocker dans la variable "Age". (Utiliser la fonction DATE(), qui donne la date actuelle)

9. Créer un format pour la variable "Sexe" afin d'avoir des modalités "Homme" et "Femme", puis l'appliquer. (PROC FORMAT)

10. Créer une variable "Mention" en suivant la règle d'attribution des mentions au baccalauréat. (Passer par un IF, veiller à formater la variable "Mention" pour que l'ensemble du texte s'affiche)

11. Afficher les tables pour chaque valeur de BAC. (Regarder du côté de PROC PRINT, en veillant à trier les donner au préalable selon la variable BAC via la PROC SORT)

Exercice 3 :

1. Exécuter le programme suivant plusieurs fois (2 fois ou plus). Que fait le programme ?

```
%let N=100;
data exemple_data(keep=a b c);
do i=1 to &N;
a=rand('BERNOULLI',0.6);
b=rand('BINOMIAL',0.1,20);
c=rand('EXPONENTIAL');
output;
end;
run;
```

2. Reprenez le programme précédent et placer la ligne de code : call streaminit(1); avant le début de la boucle et exécutez le programme plusieurs fois. Que remarquez-vous ?

3. La table de données que vous avez créée précédemment s'appelle exemple_data. Créer une table de données avec le nom IMC. Cette table doit contenir 1000 lignes et trois colonnes : Individu Poids et Taille.

La colonne Individu doit contenir le nom des observations (individu1, individu2, individu3, …, individu1000)

Aide : la fonction cats() de sas permet de concaténer des nombres et des strings

La colonne Poids doit attribuer une masse (en kilogramme) à chaque individu de façon aléatoire. La masse des individus doit obéir à une loi uniforme d'intervalle [55,90]. Attention : la colonne Poids ne doit pas contenir de nombre à virgule (prenez la partie entière de la masse).

Aide : pour obtenir la partie entière d'un nombre vous pouvez utiliser la fonction floor()

La colonne Taille doit être construite de la façon suivante : si un individu pèse m=60 kg sa taille doit être donnée par T=100+m+ (un nombre aléatoire compris entre -5 et 5). Le nombre aléatoire choisit entre -5 et 5 doit obéir à une loi uniforme. Attention : ici aussi la colonne taille ne doit pas contenir de nombre à virgule.
La table de données que vous devez obtenir doit ressembler à :

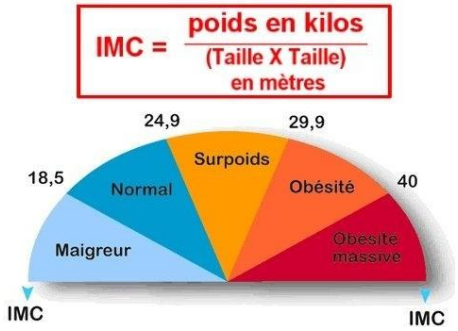
	Individu	Poids	Taille
1	individu1	75	170
2	individu2	57	155
3	individu3	66	164
4	individu4	66	162
5	individu5	56	151
6	individu6	87	184
7	individu7	71	175
8	individu8	81	184
9	individu9	66	169

Aides supplémentaires :

Nombre aléatoire :
http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a001466748.htm

4. Modifier le programme précédent afin d'ajouter une colonne imc à la table IMC.
   La formule de l'imc est donnée dans la figure suivante :

IMC = Indice de Masse Corporelle



Attention dans cette formule, la taille est exprimée en mètre.

5. Dans un autre programme (sans créer de boucle et en utilisant les conditions IF, ELSE IF, ELSE) créer une colonne diagnostic (Maigreur, Normal, Surpoids, Obésité, Obésité massive) en fonction de l'imc.

La table finale doit ressembler à :

	Individu	Poids ▲	Taille	imc	diagnostic
1	individu428	55	153	23	Surpoids
2	individu270	55	155	22	Surpoids
3	individu353	55	157	22	Surpoids
4	individu859	55	152	23	Surpoids
5	individu10	55	154	23	Surpoids
6	individu557	55	151	24	Surpoids
7	individu446	55	154	23	Surpoids
8	individu621	55	155	22	Surpoids
9	individu856	55	153	23	Surpoids
10	individu433	55	157	22	Surpoids
11	individu731	55	152	23	Surpoids
12	individu336	55	150	24	Surpoids
13	individu105	55	159	21	Surpoids
14	individu569	55	159	21	Surpoids