



Devoir sous SAS

MASTER 1 MIASHS parcours MQME

Encadrant : BASSANE Aladji

Keroudine BELLADJO

25 février 2023

Partie 1

L'objectif de cette partie est tout d'abord de reconstituer le fichier Train qui est une partie du célèbre fichier Titanic. Ce fichier est divisé en trois sous-fichiers que je vous propose de reconstituer et ensuite d'effectuer une étude statistique du fichier en entier.

Etape 1 : Importation des données

Tout d'abord on a créé une bibliothèque qu'on a nommé **DMBIB** dans laquelle on va importer nos données. Nous allons travailler avec cette bibliothèque tout au long de cette partie.

1) Ecrivez un script qui lit les trois fichiers de `train_url1`, `train_url2` et `train_url3`.

Importation et la lecture du fichiers de train_url1 :

```
/* Fichier source : train_url1.xlsx */
/* Chemin source : /home/u63203471/DM */
/* Code généré le : 20/02/2023 17:14 */

%web_drop_table(DMBIB.train_url1);
FILENAME REFFILE '/home/u63203471/DM/train_url1.xlsx';
PROC IMPORT DATAFILE=REFFILE
  DBMS=XLSX
  OUT=DMBIB.train_url1;
  GETNAMES=YES;
RUN;
PROC CONTENTS DATA=DMBIB.train_url1; RUN;
%web_open_table(DMBIB.train_url1);
/* affichage du train_url1*/
proc print data=dmbib.train_url1;
Run;
```

Importation et la lecture du fichiers de train_url2 :

```
/* Fichier source : train_url2.xlsx */
/* Chemin source : /home/u63203471/DM */
%web_drop_table(DMBIB.train_url2);
FILENAME REFFILE '/home/u63203471/DM/train_url2.xlsx';
PROC IMPORT DATAFILE=REFFILE
  DBMS=XLSX
  OUT=DMBIB.train_url2;
  GETNAMES=YES;
RUN;
PROC CONTENTS DATA=DMBIB.train_url2; RUN;
%web_open_table(DMBIB.train_url2);
/* affichage du train_url2*/
proc print data=dmbib.train_url2;
Run;
```

Importation et la lecture du fichiers de train_url3 :

```
/* Fichier source : train_url3.xlsx */
/* Chemin source : /home/u63203471/DM */
%web_drop_table(DMBIB.train_url3);
FILENAME REFFILE '/home/u63203471/DM/train_url3.xlsx';
PROC IMPORT DATAFILE=REFFILE
  DBMS=XLSX
  OUT=DMBIB.train_url3;
  GETNAMES=YES;
RUN;
PROC CONTENTS DATA=DMBIB.train_url3; RUN;
%web_open_table(DMBIB.train_url3);
/* affichage du train_url3*/
proc print data=dmbib.train_url3;
Run;
```

Concaténation :

1- Écrivez un script qui vous permet d'effectuer une concaténation entre train_url1 et train_url2 pour créer un fichier nommé train_url1_2

```
proc sort data=DMBIB.Train_url1;
by passengerId;
run;

proc sort data=DMBIB.Train_url2;
by passengerId;
run;

data DMBIB.train_url1_2;
merge DMBIB.Train_url1 DMBIB.Train_url2;
by passengerId;
run;

proc print data=DMBIB.train_url1_2;
Run;
```

2. Écrivez un script qui vous permet d'effectuer une nouvelle concaténation entre train_url1_2 et train_url3 pour créer un fichier nommé train_url.

```
data DMBIB.train_url;  
set DMBIB.train_url1_2 DMBIB.Train_url3;  
run;  
  
proc print data=DMBIB.train_url;  
Run;
```

Le nombre d'observation de La table train_url créée:

```
proc sql;  
select count(*) as Nombre_d_observations from DMBIB.train_url;  
run;
```

Nombre_d_observations
884

On a donc 884 observations dans la base de données train_url qu'on vient de créer

Etape 2 : Analyse descriptive du fichier Train_url

1. Déterminer le nombre :

a. de survivants et le nombre de non survivant ?

```
proc sql;  
select survived,count(distinct PassengerId) as Nombre  
from DMBIB.train_url group by survived ;  
quit;
```

Survived	Nombre
0	543
1	341

0 = Non survécu

1= survécu

Donc Parmi les passagers , 341 ont survécus et 543 décédés

b. de passagers hommes et femmes ?

```
proc sql;  
select Sex,count(distinct name) as Nombre  
from DMBIB.train_url  
group by Sex;  
quit;
```

Sex	Nombre
female	312
male	572

Parmi les passagers , il y'avait 572 hommes et 312 femmes.
Il y'avait donc plus d'hommes que femmes sur le Titanic.

2. Combien y a-t-il de passagers :

a. par point d'embarquement ?

```
Proc sql;  
select Embarked,count(DISTINCT PassengerId) as Nombre  
from DMBIB.train_url  
group by Embarked  
ORDER BY Nombre ASC;  
Quit;
```

Embarked	Nombre
	2
Q	76
C	167
S	639

Lors de l'embarquement dans le Titanic, Il y'a eu 76 passagers qui sont passés par le point d'embarquements Q, 167 au point d'embarquements C , 639 au point d'embarquements S et 2 dont le point d'embarquements est inconnu qui sont probablement le capitaine et son adjoint

b. hommes et femmes par classe ?

```
proc sql;  
select Sex,Pclass,count(distinct PassengerId) as  
nombre_de_passagers  
from DMBIB.train_url  
group by Pclass, Sex;  
run;
```

Sex	Pclass	nombre_de_passagers
female	1	93
male	1	121
female	2	76
male	2	108
female	3	143
male	3	343

Dans le Titanic, parmi les femmes, 93 étaient dans la classe 1 , 76 dans la classe 2 et 143 dans la classe 3. Par contre Parmi les hommes, 121 étaient dans la classe 1, 108 dans la classe 2 et 343 dans la classe 3. Il y'avait donc plus de femme dans classe 3 que dans les autres classes.Idem pour les hommes.

Etape 3 : Analyse descriptive avancée

1) Pour la variable âge :

a) Calculer la moyenne de la variable âge par sexe:

Conversions de la variables age en numérique:

```
data DMBIB.train_url;
  set DMBIB.train_url;
  Age2=input(Age,8.);
run;
```

Calcul de l'âge moyen par sexe:

```
proc sql;
select Sex,avg(Age2) as age_moyen
from DMBIB.train_url
group by Sex;
QUIT;
```

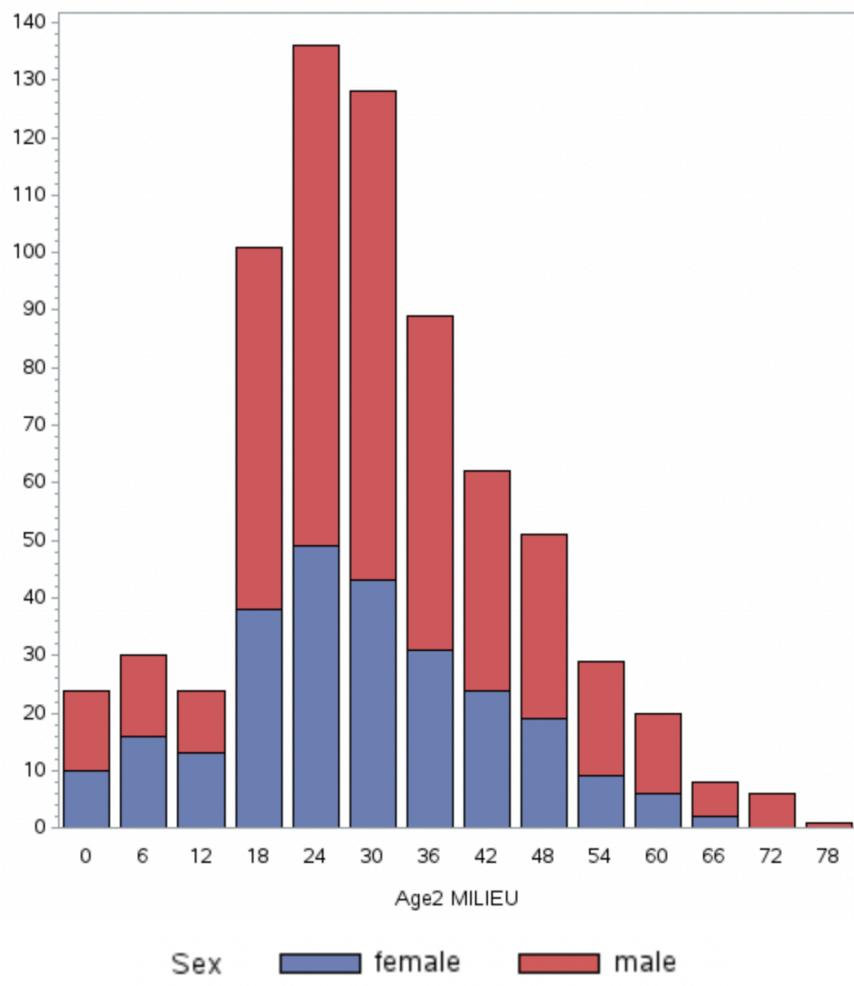
Sex	age_moyen
female	27.92692
male	30.77989

L'âge moyen des passagers de Titanic de sexe féminin est 27 ans et celui de sexes masculin est de 30 ans

b. Faire un histogramme de la variable Age par sexe. Interpréter le graphique.

```
PROC GCHART DATA=dmbib.train_url;
  VBAR Age2/ SUBGROUP=Sex;
  RUN;
QUIT;
```

Histogramme de la variable Age par sexe

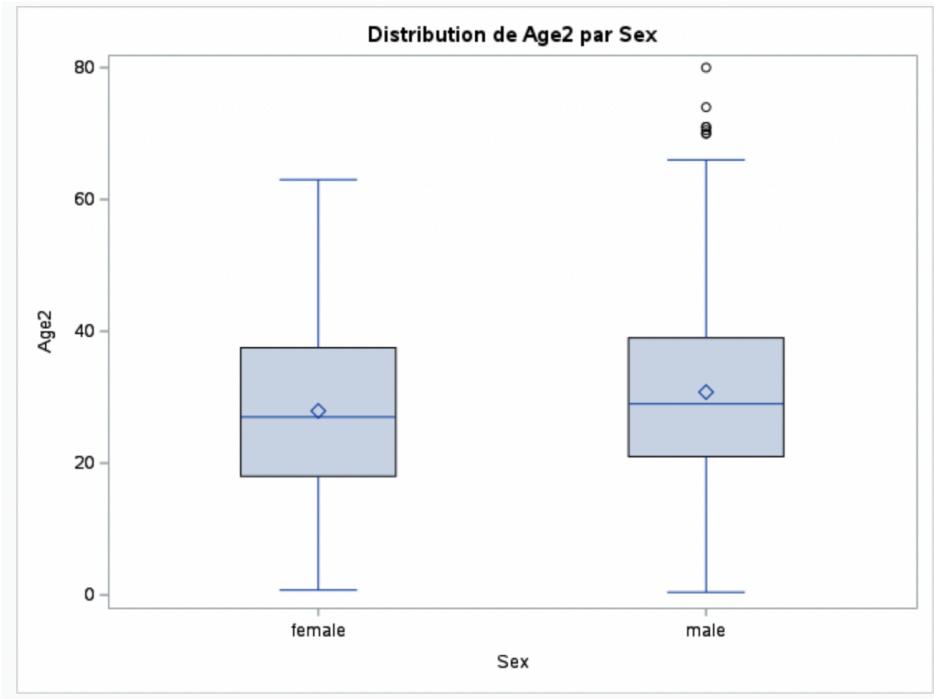


Interpretation : On constate que la majeur partie des passagers ont 24 ans. Parmi eux il y'a plus d'hommes que de femme. On remarque aussi que les passagers les plus âgées sont des hommes.

c. Faire une boîte à moustaches de la variable Age par sexe.
Interpréter le graphique

```
**ici on veut un boxplot Age par sexe**
** on trie tout d'abord les données selon sexe**
```

```
proc sort data= dmbib.train_url;
  By Sex;
run;
PROC BOXPLOT data= dmbib.train_url;
  plot age2 * Sex /
    caxes=black ctext= black
    Cboxes=black boxstyle= schematic
    idcolor=black idsymbol= dot;
run;
```



Interpretation: On remarque que la majorité des femmes ont entre 18 et 39 ans tandis que Les hommes quand à eux ont majoritairement entre 21 et 40 ans. Nous pouvons voir aussi que l'âge médiane des femmes est largement plus bas que celui des hommes. L'âge maximal des femmes est environ 63 ans et celui des hommes est de 80 ans. Les hommes ont donc âge assez élevé. Par contre les hommes et les femmes ont un âge minimale presque identique (les bébés).

2) Pour la variable Fare(Prix du ticket) :

a. Calculer la moyenne de la variables Fare par classe

Conversion de la variable Fare en numérique:

```
data DMBIB.train_url;
  set DMBIB.train_url;
  Fare_num=input(Fare,8.);
run;
```

Calcul du prix moyen du ticket par classe :

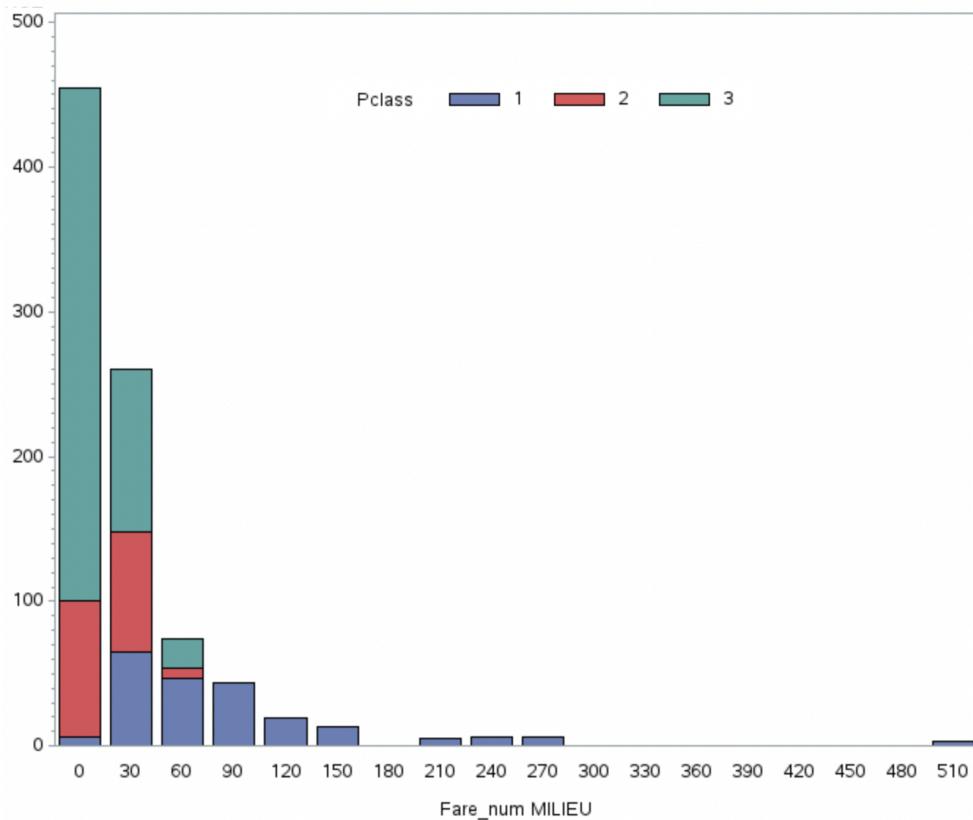
```
proc sql;
select Pclass,avg(Fare_num) as Prix_moyen_par_classe
from dmbib.train_url
group by Pclass;
quit;
```

Pclass	Prix_moyen_par_classe
1	84.09282
2	20.66218
3	13.68711

Le prix moyen du ticket pour la classe 1 est de 84.09 , 20.66 pour la classe 2 et 13.69 pour la classe 3. Les tickets de la classe 1 sont donc les plus chères.

**b. Faire un histogramme de la variable Fare par classe.
Interpréter le graphique**

```
PROC GCHART DATA=dmbib.train_url;
  VBAR Fare_num / SUBGROUP=Pclass;
  RUN;
QUIT;
```



INTERPRETATION : On remarque que la majorité des passagers étais dans la classe 3 qui correspond à la classe la moins chère.

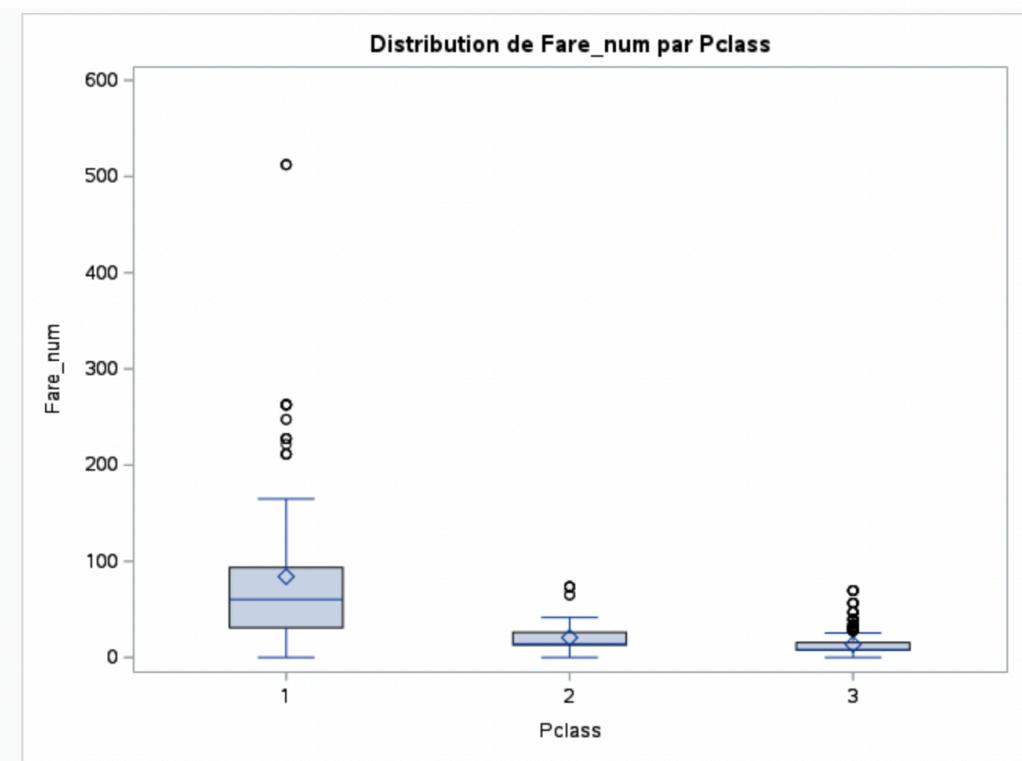
c. Faire une boîte à moustaches de la variable Fare par classe. Interpréter le graphique

```

proc sort data= dmbib.train_url;
  By Pclass;
run;

PROC BOXPLOT data= dmbib.train_url;
  plot Fare_num * Pclass /
    caxes=black ctext= black
    Cboxes=black boxstyle= schematic
    idcolor=black idsymbol= dot;
run;

```



Interpretation: On remarque que le prix des tickets varie en fonction de la classe. La classe 1 est la plus chère ensuite vient de la classe 2 puis la classe 3.

3) Déterminer :

- a. Le nombre d'hommes et de femmes par classe

```

PROC FREQ DATA = DMBIB.train_url;
TABLES Sex * Pclass;
RUN;

proc sql;
select Pclass,Sex,count(distinct PassengerId) as Nombre
from DMBIB.train_url
group by Pclass,Sex;
Run;

```

Pclass	Sex	Nombre
1	female	93
1	male	121
2	female	76
2	male	108
3	female	143
3	male	343

Dans la classe 1 de Titanic, Il y avait 93 femmes et 121 hommes. Dans la classe 2 , il y avait 78 femmes et 108 hommes et enfin dans la 3 il y avait 143 femmes et 343 hommes. Dans toutes les classe on remarque qu'il y avait plus d'hommes que de femmes.

b. La moyenne de la variable Age par classe et par sexe

```
proc sql;
select Pclass,Sex,avg(Age2) as Moyenne
from DMBIB.train_url
group by Pclass,Sex;
quit;
```

Pclass	Sex	Moyenne
1	female	34.72619
1	male	41.4342
2	female	28.72297
2	male	30.74071
3	female	21.75
3	male	26.53368

Dans la classe 1 , l'âge moyen des femmes étaient de 34 ans et celui des homme de 41 ans. Dans la classe 2 , l'âge moyen des femmes étaient de 28 ans et celui des hommes de 30 ans. Dans la classe 3, l'âge moyen des femmes étaient de 21 ans et celle des hommes de 26 ans.

On remarque que les hommes et femmes de même génération partage généralement la même classe.

4) Calculer

- a. Les pourcentages de survivants et non survivants par sexe du fichier `train_url`. Interpréter les résultats

```
PROC FREQ DATA = DMBIB.train_url;
TABLES Survived*Sex;
RUN;

PROC SQL;
  SELECT Sex,
         100 *sum(Survived=1) / count(*) as
pourcentage_survivants,
         100 *sum(Survived=0) / count(*) as
pourcentage_non_survivants
  FROM DMBIB.train_url
  GROUP BY sex;
QUIT;
```

La procédure FREQ				
Fréquence Pourcentage Pct de ligne Pct de col.	Table de Survived par Sex			
	Survived(Survived)	Sex(Sex)		
	0	79 8.94 14.55 25.32	464 52.49 85.45 81.12	543 61.43
	1	233 26.36 68.33 74.68	108 12.22 31.67 18.88	341 38.57
	Total	312 35.29	572 64.71	884 100.00

Sex	pourcentage_survivants	pourcentage_non_survivants
female	74.67949	25.32051
male	18.88112	81.11888

Parmi les hommes , seulement 18.88% ont survécus contre 81.12% décédés. Parmi les femmes seulement 25.32% décédés contre 74.67% survécus. Il y a eu donc plus de femmes survivants que d'hommes. Cela peut s'expliquer par le fait que dans les normes on sauve les femmes et les enfants d'abord.

b. Les pourcentages de survivants par point d'embarquement. Interpréter les résultats

```
PROC FREQ DATA = DMBIB.train_url;
TABLES Survived*Embarked;
RUN;
```

```
PROC SQL;
    SELECT Embarked,count(*) as
nombre_de_personnes_totales,
        100 *sum(Survived=1) / count(*) as
pourcentage_survivants
    FROM DMBIB.train_url
    GROUP BY Embarked;
QUIT;
```

La procédure FREQ						
Fréquence Pourcentage Pct de ligne Pct de col.	Table de Survived par Embarked					
	Survived(Survived)	Embarked(Embarked)				
		C	Q	S	Total	
		0	75 8.50 13.81 44.91	46 5.22 8.47 60.53	422 47.85 77.72 66.04	543 61.56
		1	92 10.43 27.14 55.09	30 3.40 8.85 39.47	217 24.60 64.01 33.96	339 38.44
	Total		167 18.93	76 8.62	639 72.45	882 100.00
Fréquence manquante = 2						

55.1% des passagers qui sont passés par le point d'embarquements C n'ont pas survécu contre 39.5% de celui de Q et 33.96% de celui de S. Les passagers qui sont passés par le point d'embarquements C ont donc moins survécu.

Embarked	nombre_de_personnes_totales	pourcentage_survivants
	2	100
C	167	55.08982
Q	76	39.47368
S	639	33.95931

c. Calculer les pourcentages de survivants par sexe et par classe. Interpréter les résultats

```
PROC SQL;
    SELECT Pclass,Sex,
           100 *sum(Survived=1) / count(*) as
    pourcentage_survivants
    FROM DMBIB.train_url
    GROUP BY Pclass,Sex;
QUIT;
```

Pclass	Sex	pourcentage_survivants
1	female	97.84946
1	male	36.36364
2	female	92.10526
2	male	15.74074
3	female	50.34965
3	male	13.70262

Dans la classe 1 , 97.85% des femmes ont survécues contre seulement 36.36% des hommes. Dans la classe 2 , 92.11% des femmes ont survécus contre seulement 15.74% des hommes. Dans la classe 3, 50% des femmes ont survécues contre seulement 13.7% des hommes. On remarque qu'il y a eu plus de mort parmi les hommes que des femmes. Les passagers de la classe 3 ont moins survécus que les passagers des autres classes.

5) Créer une variable pour différencier un adulte d'un enfant

```
data dmbib.train_url;
  set dmbib.train_url;

  if (age <=17) then individus="Enfant ";
  if (age >= 18) then individus="Adulte ";
  if (age = "") then individus="inconu ";

run;
proc print data=dmbib.train_url;
Run;
```

Obs.	PassengerId	Name	Survived	Pclass	Sex	Age	Fare	Embarked	Age2	Fare_num	individus
1	2	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	1	1	female	38	71.2833	C	38.00	71.283	Adulte
2	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	1	female	35	53.1	S	35.00	53.100	Adulte
3	12	Bonnell, Miss. Elizabeth	1	1	female	58	26.55	S	58.00	26.550	Adulte
4	32	Spencer, Mrs. William Augustus (Marie Eugenie)	1	1	female		146.5208	C	.	146.521	inconu
5	53	Harper, Mrs. Henry Sleeper (Myra Haxton)	1	1	female	49	76.7292	C	49.00	76.729	Adulte

6) Déterminer :

a. Le nombre de passagers adultes et le nombre d'enfants passagers

```
proc sql;
select individus,count(distinct PassengerId) as nombre
from dmbib.train_url
group by individus;
quit;
```

individus	nombre
Adulte	597
Enfant	112
inconu	175

Les passagers du Titanic compte 597 Adultes et 112 enfants. Les 175 autres passagers on ne connaît pas leur âge pour leur statués.

b. Le nombre d'enfants et d'adultes qui ont survécu par classe ?

```
proc sql;
select individus,Pclass,count(distinct PassengerId) as
nombre
from dmbib.train_url
where Survived=1
and individus in ("Adulte","Enfant")
group by individus,Pclass;
quit;
```

individus	Pclass	nombre
Adulte	1	110
Adulte	2	62
Adulte	3	56
Enfant	1	11
Enfant	2	21
Enfant	3	29

Dans la classe 1 , 110 adultes ont survécus contre 11 enfants.
Dans la classe 2 , 62 adultes ont survécus contre 21 enfants. Dans la classe 3 , 56 adultes ont survécus contre 29 enfants.

c. Le nombre d'enfants et d'adultes qui ont survécu par classe et par point d'embarcation ?

```
proc sql;
select
individus,Pclass,Embarked,count(distinct
PassengerId) as nombre
from dmbib.train_url
where Survived=1
and individus in ("Adulte","Enfant")
group by individus,Pclass,Embarked;
quit;
```

individus	Pclass	Embarked	nombre
Adulte	1		2
Adulte	1	C	49
Adulte	1	Q	1
Adulte	1	S	58
Adulte	2	C	4
Adulte	2	Q	1
Adulte	2	S	57
Adulte	3	C	7
Adulte	3	Q	3
Adulte	3	S	46
Enfant	1	C	3
Enfant	1	S	8
Enfant	2	C	4
Enfant	2	S	17
Enfant	3	C	11
Enfant	3	Q	3
Enfant	3	S	15

Partie 2 : SQL en langage SAS

Le but de cette partie est de réécrire les procédures en langage uniquement. Vous n'êtes pas autorisé à utiliser la procédure sql.

1) Restriction : clause where, and, or et opérateurs classiques*/

/ 1 */*

Code sql

```
PROC SQL;
  SELECT *
    FROM TD.CLIENT
   WHERE cpcli <> 57500;
QUIT;
```

code SAS

```
Data Tab1;
  set TD.CLIENT (where=(cpcli <> 57500));
Run;
proc print data =Tab1;
Run;
```

codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age
5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming	H	18
10	David	Collague	14 rue Utrillo	57600	Forbach	H	22
12	Julie	Duart	19 boulevard de Belfort	5900		F	18

Sql

Obs.	codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age
1	5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming	H	18
2	10	David	Collague	14 rue Utrillo	57600	Forbach	H	22
3	12	Julie	Duart	19 boulevard de Belfort	5900		F	18

SAS

```

/* 2*/
/* sql*/

PROC SQL;
  SELECT *
    FROM TD.CLIENT
   WHERE villecli= "Saint-Avold"
     AND sexe= "H"
     AND age < 23;
QUIT;

```

```

/*SAS*/
data Tab2;
  set TD.CLIENT
    (where=(villecli= "Saint-Avold"
      and sexe= "H"
      AND age < 23));
run;
proc print data =Tab2;
Run;

```

codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age
3	Roger	Botas	5 place du marche	57500	Saint-Avold	H	21

Sql

Obs.	codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age
1	3	Roger	Botas	5 place du marche	57500	Saint-Avold	H	21

SAS

2. Restriction: where, upper, lower, between, is null

```

/* 3*/
/* sql*/

PROC SQL;
  SELECT *
    FROM TD.CLIENT
   WHERE age BETWEEN 18 AND 21;
QUIT;

```

```

/* sas */

data Tab3;
  set TD.CLIENT
    (where=(age>=18 AND age <= 21));
run;
proc print data =Tab3;
Run;

```

codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age
3	Roger	Botas	5 place du marche	57500	Saint-Avold	H	21
5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming	H	18
12	Julie	Duart	19 boulevard de Belfort	5900		F	18

SQL

Obs.	codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age
1	3	Roger	Botas	5 place du marche	57500	Saint-Avold	H	21
2	5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming	H	18
3	12	Julie	Duart	19 boulevard de Belfort	5900		F	18

SAS

```

/* 4*/
/* sql*/

PROC SQL;
  SELECT *
    FROM TD.CLIENT
   WHERE sexe IN ( 'H' , 'F' );
QUIT;

/* sas*/

data Tab4;
  set TD.CLIENT
    (where=(Sexe='H' or Sexe = 'F'));
run;
proc print data =Tab4;
Run;

```

codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age
3	Roger	Botas	5 place du marche	57500	Saint-Avold	H	21
5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming	H	18
10	David	Collague	14 rue Utrillo	57600	Forbach	H	22
12	Julie	Duart	19 boulevard de Belfort	5900		F	18

SQL

Obs.	codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age
1	3	Roger	Botas	5 place du marche	57500	Saint-Avold	H	21
2	5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming	H	18
3	10	David	Collague	14 rue Utrillo	57600	Forbach	H	22
4	12	Julie	Duart	19 boulevard de Belfort	5900		F	18

SAS

3. Restriction: where, in, like

```
/* 5*/
/* sql*/
```

```
PROC SQL;
      SELECT *
FROM TD.CLIENT
WHERE Nomcli LIKE 'L%';
QUIT;
```

```
/* sas*/
data Tab5;
  set TD.CLIENT
    (where=(SUBSTR(Nomcli,1,1)="L"));
run;
proc print data =Tab5;
Run;
```

codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age
5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming	H	18

SQL

Obs.	codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age
1	5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming	H	18

SAS

```

/* 6*/
/* sql*/

PROC SQL;
SELECT DISTINCT Villecli
  FROM TD.CLIENT;
QUIT;

/* sas*/

proc sort data = TD.CLIENT nodupkey;
by Villecli;
run;
proc print data =TD.CLIENT;
var Villecli;
Run;

```

SQL

villecli
Forbach
Freyming
Saint-Avold

SAS

Obs.	villecli
1	
2	Forbach
3	Freyming
4	Saint-Avold

4. Gestion des doublons : Distinct et renommage

```

/* 7*/
/* sql*/

PROC SQL;
      SELECT DISTINCT Villecli as ville, sexe FROM
TD.CLIENT;
QUIT;

```

```

/* sas*/
proc sort data = TD.CLIENT OUT=Tab7 nodupkey;
by Villecli Sexe;
run;
proc print data =Tab7 (rename=(Villecli=ville));
var Ville Sexe;
Run;

```

SQL

ville	Sexe
	F
Forbach	H
Freyming	H
Saint-Avold	H

SAS

Obs.	ville	Sexe
1		F
2	Forbach	H
3	Freyming	H
4	Saint-Avold	H

5. Les jointures : INNER JOIN - LEFT JOIN

```

/* 8*/
/* sql*/

PROC SQL;
      SELECT *
      FROM TD.CLIENT cl
INNER JOIN TD.location lo
ON cl.codecli=lo.codecli
INNER JOIN TD.film fi
ON lo.codefilm =fi.codefilm
      WHERE cl.cpcli=57800;
QUIT;

/* sas*/
/*trie de la table client*/
proc sort data=TD.CLIENT;
by codecli;
run;

```

```

/*trie de la table location*/
proc sort data=TD.location;
by codecli;
run;

/*jointure de la table client et location par la methode
merge*/
data Tab8_1;
merge TD.CLIENT (in=cli) TD.location (in=loc);
by codecli;
if cli and loc;
run;

/*trie de la table Tab8_1 obtenue de la jointure*/
proc sort data=Tab8_1;
by codefilm;
run;

/*trie de la table film*/
proc sort data=TD.film;
by codefilm;
run;

/*jointure de la table Tab8_1 et film par la methode merge*/
data Tab8_2;
merge Tab8_1 (in=tab) TD.film (in=fi);
by codefilm;
if tab and fi;
run;

Data Tab8_3;
    set Tab8_2 (where=(cpcli=57800));
Run;

proc print data= Tab8_3;
run;

```

codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age	codecli	codefilm	datedebut	duree	codefilm	nomfilm
5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming	H	18	5	2	25/04/2013	3	2	Bernie

SQL

Obs.	codecli	prenomcli	nomcli	ruecli	cpcli	villecli	Sexe	Age	codefilm	datedebut	duree	codefilm	nomfilm
1	5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming	H	18	2	25/04/2013	3	2	Bernie

SAS

```

/* 9*/
/* sql*/

PROC SQL;
    SELECT *
        FROM TD.client cl
    LEFT JOIN TD.location Lo
        ON cl.codecli=lo.codecli
INNER JOIN TD.film fi
        ON lo.codefilm =fi.codefilm
    WHERE fi.codefilm IN (1,5,7);
QUIT;

/* sas*/

PROC SORT DATA=TD.client;
    BY codecli;
RUN;

DATA Tab9_1;
    MERGE TD.client(in=cli) TD.location(in=loc);
    BY codecli;
RUN;

proc sort DATA=Tab9_1;
    BY codecli;
RUN;

DATA Tab9_2;
    MERGE Tab9_1(in=join) TD.film(in=fm);
    BY codefilm;
    if join;
RUN;

DATA Tab9_3;
    SET Tab9_2;
    IF codefilm in (1, 5, 7);
RUN;

PROC PRINT DATA=Tab9_3;
RUN;

```

6. Requêtes et calculs arithmétiques

```

/* 10*/
/* sql*/

PROC SQL;
  SELECT *
    FROM TD.produit
   WHERE (CodeCateg=1 AND PrixUnit > 50)
      OR (CodeCateg=3 AND PrixUnit < 90);
QUIT;

/* sas*/
Data Tab10;
  set TD.produit (where=(CodeCateg=1 and PrixUnit > 50
or CodeCateg=3 AND PrixUnit < 90));
Run;
proc print data=Tab10;
Run;

```

Refprod	Nomprod	NoFour	CodeCateg	QteParUnit	PrixUnit	UnitesStock	UnitesCom	NiveauReap	Indisponible
1	Chai	1	1	10 boites x 20 sacs	90	39	0	10	0
2	Chang	1	1	24 bouteilles (1 litre)	95	17	40	25	0
16	Pavlova	7	3	32 boites (500 g)	87.25	29	0	10	0
19	Teatime Chocolate Biscuits	8	3	10 boites x 12 pieces	46	25	0	5	0
21	Sir Rodneys Scones	8	3	24 cartons x 4 pieces	50	3	40	5	0

SQL

Obs.	Refprod	Nomprod	NoFour	CodeCateg	QteParUnit	PrixUnit	UnitesStock	UnitesCom	NiveauReap	Indisponible
1	1	Chai	1	1	10 boites x 20 sacs	90	39	0	10	0
2	2	Chang	1	1	24 bouteilles (1 litre)	95	17	40	25	0
3	16	Pavlova	7	3	32 boites (500 g)	87.25	29	0	10	0
4	19	Teatime Chocolate Biscuits	8	3	10 boites x 12 pieces	46	25	0	5	0
5	21	Sir Rodneys Scones	8	3	24 cartons x 4 pieces	50	3	40	5	0

SAS

```

/* 11*/
/* sql*/

PROC SQL;
  SELECT RefProd,  PrixUnit *(UnitesStock-5)
  as soustraction
    FROM TD.produit
   WHERE UnitesStock >= 5;
QUIT;

```

```

/* sas */

Data Tab11_1;
    set TD.produit (where=(UnitesStock >= 5));
Run;

data Tab11_2;
    set Tab11_1;
    Soustraction=PrixUnit *(UnitesStock-5);
run;

data Tab11_3;
    set Tab11_2 (keep=RefProd soustraction);
run;

proc print data=Tab11_3;
Run;

```

SQL

Refprod	soustraction
1	3060
2	1140
3	400
4	5280
6	14375
7	1500
8	200
9	11640
10	4030
11	1785
12	15390
13	570
14	3487.5
15	2635
16	2094
18	11562.5
19	920
20	14175
22	10395
23	2520
24	337.5

SAS

Obs.	Refprod	Soustraction
1	1	3060.0
2	2	1140.0
3	3	400.0
4	4	5280.0
5	6	14375.0
6	7	1500.0
7	8	200.0
8	9	11640.0
9	10	4030.0
10	11	1785.0
11	12	15390.0
12	13	570.0
13	14	3487.5
14	15	2635.0
15	16	2094.0
16	18	11562.5
17	19	920.0
18	20	14175.0
19	22	10395.0
20	23	2520.0
21	24	337.5

7. Traitement conditionnel: CASE WHEN

```
/* 12 */
/* sql */

PROC SQL;
  SELECT Refprod, Nomprod, PrixUnit,
CASE
    WHEN PrixUnit<=50 THEN "Bas prix"
    WHEN 50 < PrixUnit <=90 THEN "Prix moyens"
    ELSE "Produits de luxe"
  END AS Gamme
  FROM TD.produit ;
QUIT;

/* sas */

data Tab12_1;
  set TD.produit (keep=Refprod Nomprod PrixUnit);
run;

Data Tab12_2;
  Set Tab12_1;
  format Gamme $32.;

  IF PrixUnit<=50 Then
    Gamme="Bas prix";

  IF 50 < PrixUnit <=90 Then
    Gamme="Prix moyens";

  IF PrixUnit>90 Then
    Gamme="Produits de luxe";
Run;

proc print data=Tab12_2;
run;
```

SQL

Refprod	Nomprod	PrixUnit	Gamme
1	Chai	90	Prix moyens
2	Chang	95	Produits de luxe
3	Aniseed Syrup	50	Bas prix
4	Chef Antons Cajun Seasoning	110	Produits de luxe
5	Chef Antons Gumbo Mix	106.75	Produits de luxe
6	Grandmas Boysenberry Spread	125	Produits de luxe
7	Uncle Bobs Organic Dried Pears	150	Produits de luxe
8	Northwoods Cranberry Sauce	200	Produits de luxe
9	Mishi Kobe Niku	485	Produits de luxe
10	Ikura	155	Produits de luxe
11	Queso Cabrales	105	Produits de luxe
12	Queso Manchego La Pastora	190	Produits de luxe
13	Konbu	30	Bas prix
14	Tofu	116.25	Produits de luxe
15	Genen Shouyu	77.5	Prix moyens
16	Pavlova	87.25	Prix moyens
17	Alice Mutton	195	Produits de luxe
18	Carnarvon Tigers	312.5	Produits de luxe
19	Teatime Chocolate Biscuits	46	Bas prix
20	Sir Rodneys Marmalade	405	Produits de luxe
21	Sir Rodneys Scones	50	Bas prix
22	Gustafs Knackebrod	105	Produits de luxe
23	Tunnbrod	45	Bas prix
24	GuaraneFantastica	22.5	Bas prix

SAS

Obs.	Refprod	Nomprod	PrixUnit	Gamme
1	1	Chai	90	Prix moyens
2	2	Chang	95	Produits de luxe
3	3	Aniseed Syrup	50	Bas prix
4	4	Chef Antons Cajun Seasoning	110	Produits de luxe
5	5	Chef Antons Gumbo Mix	106.75	Produits de luxe
6	6	Grandmas Boysenberry Spread	125	Produits de luxe
7	7	Uncle Bobs Organic Dried Pears	150	Produits de luxe
8	8	Northwoods Cranberry Sauce	200	Produits de luxe
9	9	Mishi Kobe Niku	485	Produits de luxe
10	10	Ikura	155	Produits de luxe
11	11	Queso Cabrales	105	Produits de luxe
12	12	Queso Manchego La Pastora	190	Produits de luxe
13	13	Konbu	30	Bas prix
14	14	Tofu	116.25	Produits de luxe
15	15	Genen Shouyu	77.5	Prix moyens
16	16	Pavlova	87.25	Prix moyens
17	17	Alice Mutton	195	Produits de luxe
18	18	Carnarvon Tigers	312.5	Produits de luxe
19	19	Teatime Chocolate Biscuits	46	Bas prix
20	20	Sir Rodneys Marmalade	405	Produits de luxe
21	21	Sir Rodneys Scones	50	Bas prix
22	22	Gustafs Knackebrod	105	Produits de luxe
23	23	Tunnbrod	45	Bas prix
24	24	GuaraneFantastica	22.5	Bas prix

```

/* 13*/
/* sql*/

PROC SQL;
    SELECT Refprod, UnitesStock, UnitesCom,
           NiveauReap,
    CASE WHEN UnitesCom > 40 THEN "Déjà commandé"
         WHEN UnitesCom < NiveauReap THEN "A Commander "
         WHEN UnitesCom = 0 THEN "N'est plus en stock "
         ELSE " Disponible "
    END AS Informations
    FROM TD.produit;
QUIT;

/* sas*/
data Tab13_1;
    set TD.produit (keep=Refprod UnitesStock UnitesCom
NiveauReap);
run;

Data Tab13_2;;
    Set Tab13_1;
    select;

        when (UnitesCom > 40) Informations="Déjà commandé";

        when (UnitesCom < NiveauReap) Informations="A
Commander ";

        when (UnitesCom = 0) Informations="N'est plus en
stock ";

        otherwise Informations=" Disponible ";
    end;
Run;

proc print data=Tab13_2;
run;

```

SQL

Refprod	UnitesStock	UnitesCom	NiveauReap	Informations
1	39	0	10	A Commander
2	17	40	25	Disponible
3	13	70	25	Déjà commandé
4	53	0	0	N'est plus en stock
5	0	0	0	N'est plus en stock
6	120	0	25	A Commander
7	15	0	10	A Commander
8	6	0	0	N'est plus en stock
9	29	0	0	N'est plus en stock
10	31	0	0	N'est plus en stock
11	22	30	30	Disponible
12	86	0	0	N'est plus en stock
13	24	0	5	A Commander
14	35	0	0	N'est plus en stock
15	39	0	5	A Commander
16	29	0	10	A Commander
17	0	0	0	N'est plus en stock
18	42	0	0	N'est plus en stock
19	25	0	5	A Commander
20	40	0	0	N'est plus en stock
21	3	40	5	Disponible
22	104	0	25	A Commander
23	61	0	25	A Commander
24	20	0	0	N'est plus en stock

SAS

Obs.	Refprod	UnitesStock	UnitesCom	NiveauReap	Informations
1	1	39	0	10	A Commander
2	2	17	40	25	Disponible
3	3	13	70	25	Déjà commandé
4	4	53	0	0	N'est plus en st
5	5	0	0	0	N'est plus en st
6	6	120	0	25	A Commander
7	7	15	0	10	A Commander
8	8	6	0	0	N'est plus en st
9	9	29	0	0	N'est plus en st
10	10	31	0	0	N'est plus en st
11	11	22	30	30	Disponible
12	12	86	0	0	N'est plus en st
13	13	24	0	5	A Commander
14	14	35	0	0	N'est plus en st
15	15	39	0	5	A Commander
16	16	29	0	10	A Commander
17	17	0	0	0	N'est plus en st
18	18	42	0	0	N'est plus en st
19	19	25	0	5	A Commander
20	20	40	0	0	N'est plus en st
21	21	3	40	5	Disponible
22	22	104	0	25	A Commander
23	23	61	0	25	A Commander
24	24	20	0	0	N'est plus en st

8. Agrégation

```
/* 14*/
/* sql*/
PROC SQL;
SELECT NoFour,
       COUNT(*) AS Nombre_de_produits,
       ROUND(AVG(PrixUnit),.001) AS PrixMoyen,
       MIN(PrixUnit) AS PrixMinimum,
       MAX(PrixUnit) AS PrixMaximum
FROM TD.produit
GROUP BY NoFour
ORDER BY NoFour;
QUIT;

/* sas*/

data Tab14_1 ;
set TD.produit;
run;

PROC SUMMARY DATA=Tab14_1 NWAY;
  CLASS NoFour;
  VAR PrixUnit;
  OUTPUT OUT=Tab14_2 (DROP=_type_ _freq_)
    N=Nombre_de_produits
    MEAN=PrixMoyen
    MIN=PrixMinimum
    MAX=PrixMaximum;
RUN;

PROC SORT DATA=Tab14_2;
  BY NoFour;
RUN;

PROC PRINT DATA=Tab14_2;
  VAR NoFour Nombre_de_produits PrixMoyen
  PrixMinimum PrixMaximum;
RUN;
```

SQL

NoFour	Nombre_de_produits	PrixMoyen	PrixMinimum	PrixMaximum
1	3	78.333	50	95
2	2	108.375	106.75	110
3	3	158.333	125	200
4	2	320	155	485
5	2	147.5	105	190
6	3	74.583	30	116.25
7	3	198.25	87.25	312.5
8	3	167	46	405
9	2	75	45	105
10	1	22.5	22.5	22.5

SAS

Obs.	NoFour	Nombre_de_produits	PrixMoyen	PrixMinimum	PrixMaximum
2	1	3	78.333333333	50	95
3	2	2	108.375	106.75	110
4	3	3	158.3333333	125	200
5	4	2	320	155	485
6	5	2	147.5	105	190
7	6	3	74.583333333	30	116.25
8	7	3	198.25	87.25	312.5
9	8	3	167	46	405
10	9	2	75	45	105
11	10	1	22.5	22.5	22.5

```

/* 15*/
/* sql*/
PROC SQL;
SELECT NoFour,
       COUNT(*) AS Nombre_de_produits,
       ROUND(AVG(PrixUnit)) AS PrixMoyen,
       MIN(PrixUnit) AS PrixMinimum,
       MAX(PrixUnit) AS PrixMaximum
FROM TD.produit
   GROUP BY NoFour
HAVING Nombre_de_produits < 2
ORDER BY NoFour;
QUIT;

/* sas*/
data Tab15_1 ;
set TD.produit;
run;
PROC SUMMARY DATA=Tab15_1 NWAY;
  CLASS NoFour;
  VAR PrixUnit;
  OUTPUT OUT=Tab15_2
    (DROP=_type_ _freq_)
    N=Nombre_de_produits
    MEAN=PrixMoyen
    MIN=PrixMinimum
    MAX=PrixMaximum;
RUN;

DATA Tab15_3;
  SET Tab15_2;
  IF Nombre_de_produits < 2;
RUN;

PROC PRINT DATA=Tab15_3;
  VAR NoFour Nombre_de_produits PrixMoyen PrixMinimum
      PrixMaximum;
RUN;

```

NoFour	Nombre_de_produits	PrixMoyen	PrixMinimum	PrixMaximum
10	1	23	22.5	22.5

SQL

Obs.	NoFour	Nombre_de_produits	PrixMoyen	PrixMinimum	PrixMaximum
11	10	1	22.5	22.5	22.5

SAS