Conservatoire National des Arts et Métiers
Pascal AUREGAN

Rapport de projet

NFE211

Test de chaine décisionnelle sur cas simple.
Base opérationnelle en fichier et PostgresSQL
ETL avec TALEND Data Integration,
Reporting avec SAS

le cnam

Table des matières

1 Introduction **4**

2 Jeu de données de la base OLTP **5**

2.1 Modèle relationnel 5

2.1.1 Présentation 5

3 TALEND Open Studio for Data Integration 5.6 **7**

3.1 Création du modèle OLAP 7

3.1.1 Question à répondre 7

3.1.2 Modèle ROLAP 7

3.2 Intégration avec TALEND 9

3.2.1 Présentation 9

3.2.2 Méthodologie utilisée 9

3.2.3 Configuration des sources de données 10

3.2.4 Dimension Boutique 10

3.2.5 Dimension Livre 11

3.2.6 Dimension Client 11

3.2.7 Dimension Temps 12

3.2.8 Table de faits vente 12

3.2.9 Conclusion sur l'utilisation de TALEND 13

4 Reporting **14**

4.1 SAS 14

4.1.1 Présentation 14

4.1.2 Méthodologie utilisée 14

4.1.3 Conclusions sur l'utilisation de SAS 15

5 Conclusion **16**

A Annexes **17**

A.1 Code R ayant généré les données 17

A.2 Routine java toTrancheDage.java 20

A.3 import.sas 20

A.4 reporting.sas 23

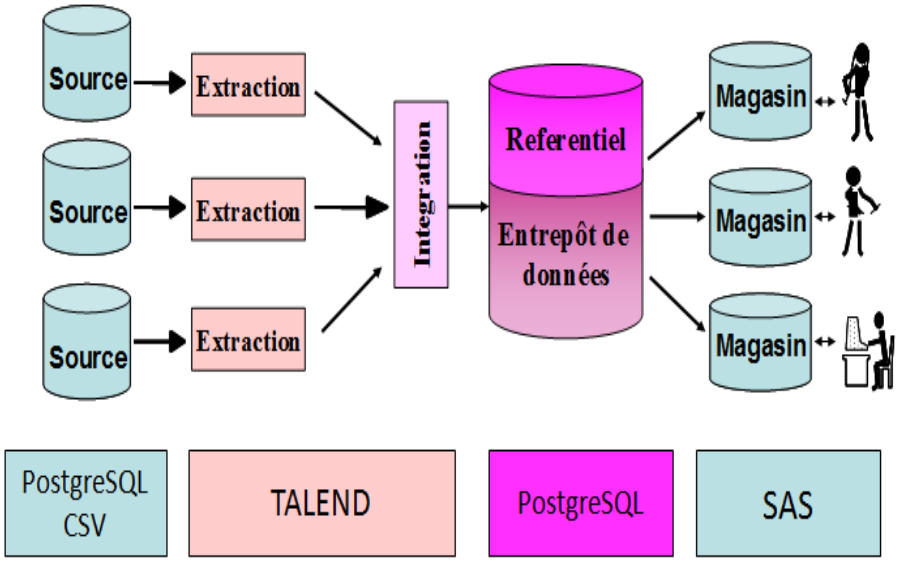
A.5 report sas 29

1 Introduction

Le but de se projet est de mettre en place une chaine décisionnelle. L'objectif de ce projet n'étant pas de pousser dans ses retranchements les outils de la chaine décisionnelle, il a été décidé de créer de toute pièce un jeu de données. Le thème de ce jeu de données est l'achat de BD comme dans le cours. La base opérationnelle décrit les factures d'un ensemble librairies en France dont les clients sont connus. La chaine décisionnelle doit être en mesure de répondre in fine à la question : analyser les ventes de BD en France en fonction des clients, du lieu et date d'achat. La couche opérationnelle est assurée par une base de données de fichier csv et une base de données PostgresSQL.

Pour la couche ETL, il a été choisi TALEND Data Integration.

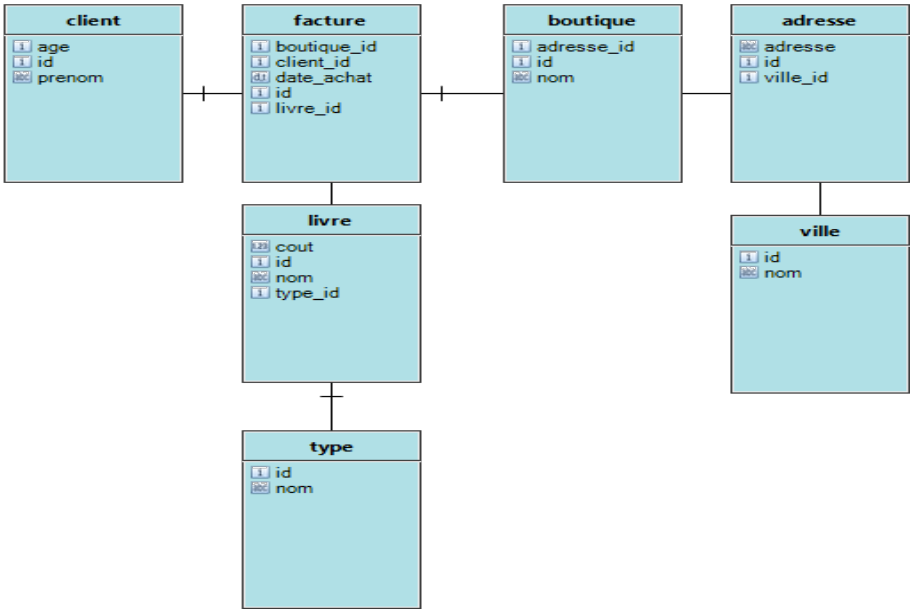
La couche reporting étudiée lors de ce projet est assurée par SAS Studio Version Etudiante.



2 Jeu de données de la base OLTP

2.1 Modèle relationnel

2.1.1 Présentation



Le modèle relationnel est un modèle normalisé.

— Une facture a été établie pour un client donné dans une boutique donnée à une date donnée et en rapport avec un livre donné.

— Le livre appartient à un type donné (Roman, BD, etc.).

— La boutique a une adresse située dans une ville

Voici combien de lignes ont chaque table :

— adresse : 26

— boutique : 25

— client : 8819

— facture : 8819

— livre : 8

— type : 3

En annexe A.1 , le code R qui a servi à générer les données dont il a été estimé que la pertinence n'avait que peu d'importance pour l'exercice. De plus, nous possédons un fichier de type csv contenant un dictionnaire de villes avec les départements associés ainsi que les régions. Voici en exemple les quatre premières lignes de ce fichier villes_def.csv

```
"ville","departement","region_name"
"Beauvais","Oise","Picardie"
"Compiegne","Oise","Picardie"
"Clermont","Oise","Picardie"
```

3 TALEND Open Studio for Data Integration 5.6

3.1 Création du modèle OLAP

3.1.1 Question à répondre

Analyse des ventes de bande dessinées d'une enseigne de librairie possédant plusieurs boutiques dans plusieurs villes.

3.1.2 Modèle ROLAP

L'implémentation ROLAP a été choisie par rapport au modèle MOLAP car même si la quantité de données était petite, il m'a semblé important d'étudier ce modèle qui paraît être très utilisé. Ne voyant rien justifier un modèle en flocon permettant de gagner de l'espace de stockage, j'ai utilisé un modèle en étoile. La table de faits est la table vente. Les dimentsions sont le temps, la boutique(lieu d'achat), le client, et le livre. L'étoile est de type transaction.

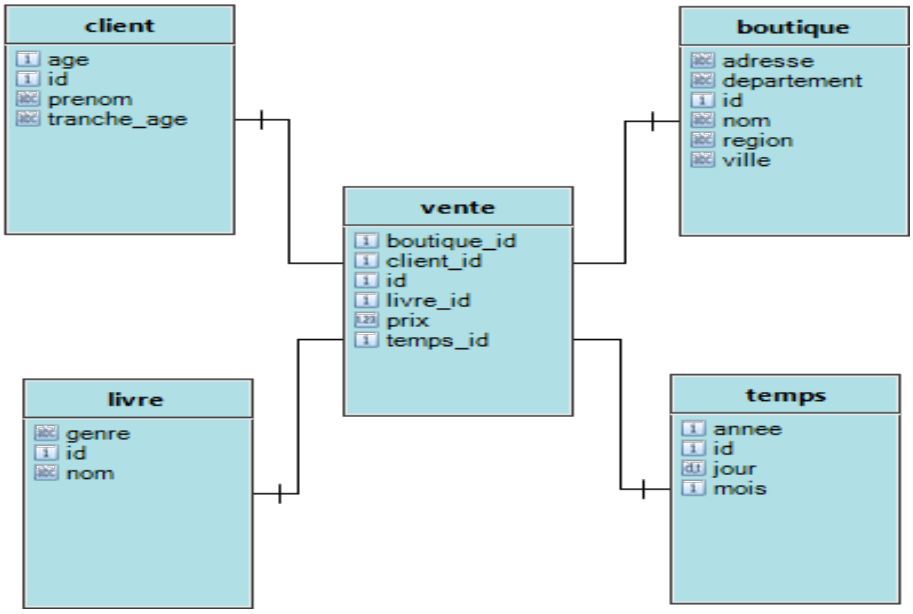


Table de faits vente

Indicateurs prix de la vente ; ce que le vente a rapporté

fonctions d'agrégat

— addition sur le prix

— moyenne sur le prix

Dimension client

— id

— prenom

— age

— tranche_age ("-12", "12-25", "35-65", "25-34", "+65")

hiérarchie age $<$ tranche_age

Dimension boutique

— id

— nom

— adresse

— departement

— ville

— region

hiérarchie adresse < ville < department < region

Dimension livre

— id
— nom
— genre

hiérarchie nom < genre

Dimension Temps

— id
— jour
— mois
— annee

hiérarchie jour < mois < annee

3.2 Intégration avec TALEND

3.2.1 Présentation

Le téléchargement se fait sur http://fr.talend.com/download/data-integration. La documentation de l'outil est disponible au même endroit dans la partie Manuels d'utilisateurs et est facilement accessible. L'outil est en fait basé sur Eclipse ce qui le rend facilement appréhendable pour ceux qui connaissent le célèbre IDE mais vient aussi avec ses défauts.

3.2.2 Méthodologie utilisée

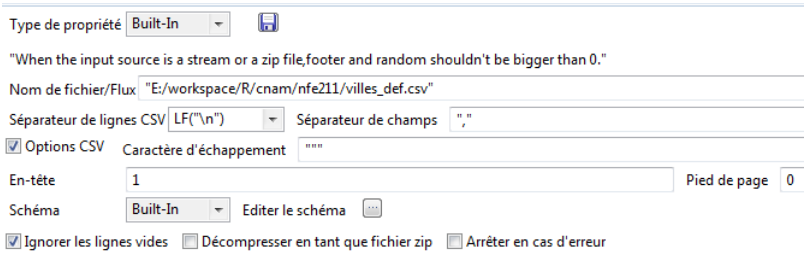
Pour chaque dimension, j'ai essayé d'avoir un cas d'usage différent :

— La dimension livre se construit par une jointure classique entre deux tables.
— La dimension client est construite en utilisant la table client suivie d'une transformation sur l'un de ses champs afin de déduire la tranche d'age à partir de l'age.
— La dimension boutique utilise des sources de données de type différents (CSV et postgres).
— La dimension temps construite à partir de la table temps subit un action sur ses données afin d'enlever les doublons et de déterminer les hiérarchies à partir d'une date.

3.2.3 Configuration des sources de données

Talend doit se connecter à la base de données postgres en lecture d'une part pour lire les données des tables de la base de données OLTP, d'autre part pour écrire en base dans la base de données de type ROLAP.

La source CSV est toute aussi facile à configurer. Il suffit de configurer le séparateur, la présence ou non d'une entête et les colonnes à prendre en compte comme le montre la figure ci-dessous :



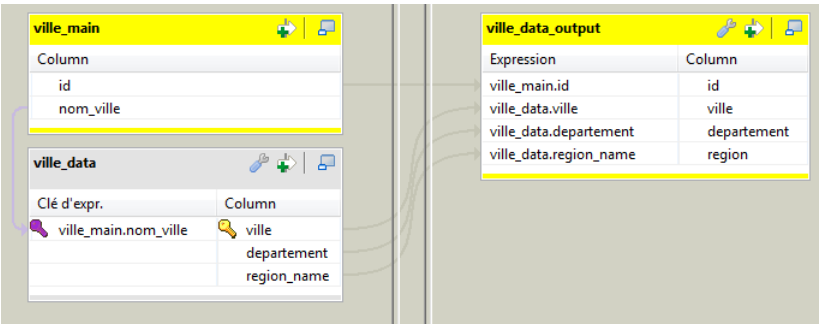
3.2.4 Dimension Boutique

Difficulté testée

La difficulté ici est de lier des données entre deux sources de données de type différents : CSV et Database.

Mise en œuvre

Les deux sources de données configurées nous permettent de faire une jointure simple avec le composant tMap. Le lien entre les entités se fait de manière intuitive sur cet écran :



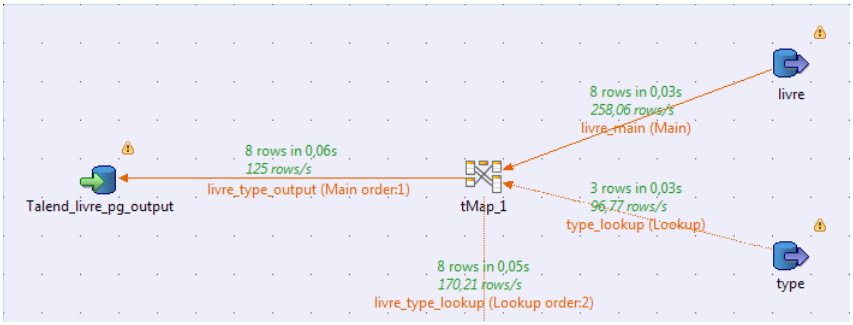
Dans la suite de ce projet nous utiliserons toujours une tMap pour faire une jointure entre deux sources de données.
Nous pouvons conclure que TALEND ne fait pas de distinction entre les différents types de données pourvu qu'elle soit composée de colonnes.

3.2.5 Dimension Livre

Difficulté testée

Aucune, jointure simple entre deux tables d'une base de données.

Mise en œuvre

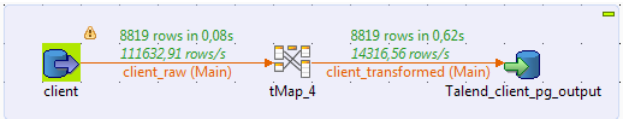


3.2.6 Dimension Client

Difficulté testée

La dimension client est construite en utilisant la table client suivie d'une transformation sur l'un de ses champs afin de déduire la tranche d'age à partir de l'age.

Mise en œuvre



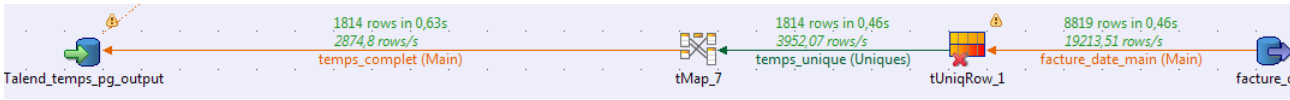
Pour transformer un age en tranche d'age, nous avons besoin de créer une routine dont le code est montré en annexe A.2 .

3.2.7 Dimension Temps

Difficulté testée

La dimension temps construite à partir de la table temps subit un action sur ses données afin d'enlever les doublons et de déterminer les hiérarchies à partir d'une date.

Mise en œuvre



Le composant tUniqRow_ que nous voyons sur ce schéma sert à enlever les doublons. C'est à dire que nous voulons une seule date dans notre dimension temps. Nous prenons donc l'ensemble des dates distinctes trouvées dans la table facture. Une fois ces dates trouvées, nous voulons déterminer les hiérarchies à partir du champs date_achat de la table facture. Nous utilisons pour cela les fonctions built-in de TALEND :

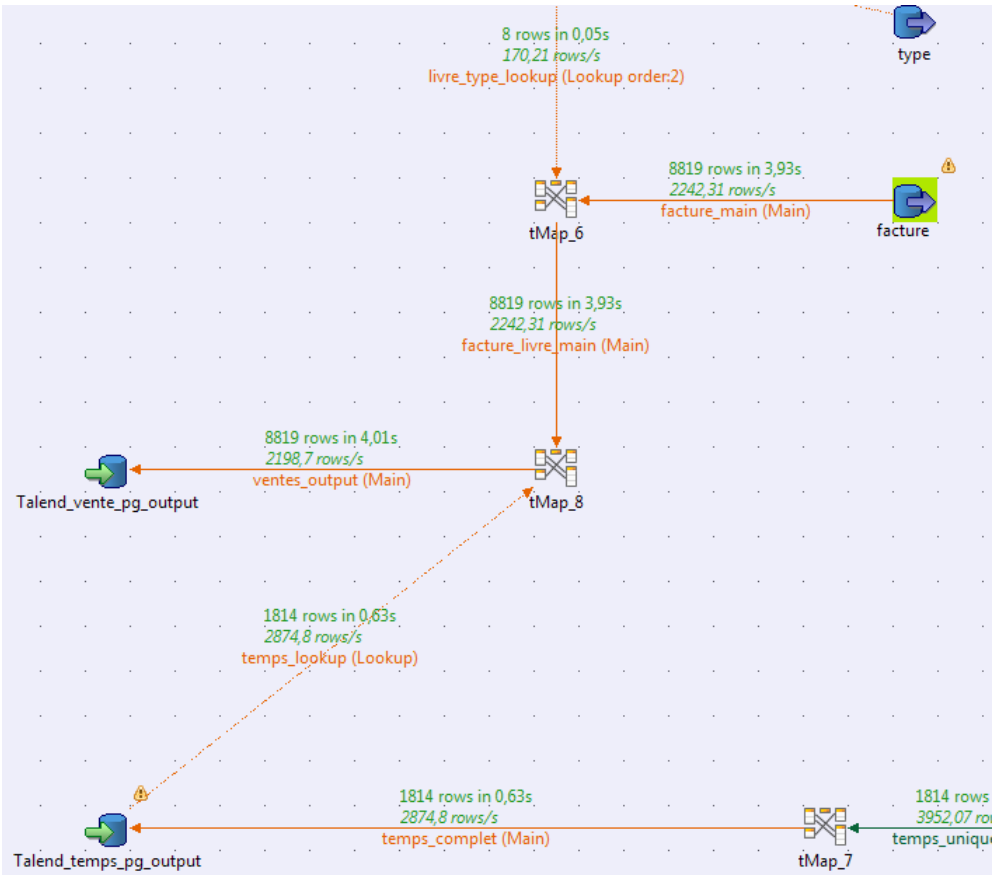
- TalendDate.getPartOfDate("MONTH", date_achat) pour déterminer le mois d'une date
- TalendDate.getPartOfDate("YEAR", date_achat)

3.2.8 Table de faits vente

Difficulté testée

Aucune, l'intégrité relationnelle des données est gérée côté base OLTP, ce qui réduit la création de la table de faits à un ensemble de jointure.

Mise en œuvre



Aucune jointure avec la table client et la table boutique n'est nécessaire puisque les relations sont prises directement de la base OLTP et qu'aucune transormation sur les clés étrangères n'est effectuée.

3.2.9 Conclusion sur l'utilisation de TALEND

Pour les cas testés qui paraissent classiques, TALEND se montre plutôt efficace. La montée en charge reste à tester. Le nombre de composant paraît énorme et je n'ai pas réussi à les utiliser tous. C'est d'ailleurs un des défauts de TALEND à mon sens. L'utilisation n'est pas très intuitive et la résolution de problème est très compliquée. Lorsqu'un problème survient, l'utilisateur est gratifié d'un NullPointerException sans plus d'information. La modification du code JAVA généré est anecdotique selon moi même si son utilité est révélée lorqu'il s'agit de débugger.

4 Reporting

4.1 SAS

4.1.1 Présentation

SAS étant soumis à licence payante, la version testée est la version University Edition. Il s'agit en fait d'un machine virtuelle sur laquelle SAS est installée en version limitée. L'accès se fait via un explorateur pointant vers SASStudio (`localhost:10080/`). Cette version limitée limite notamment l'accès aux sources de données. Cette version ne supporte donc pas l'accès à une base de données comme postgresql. Elle ne support que l'accès à des fichiers déposés dans les répertoires partagés de la machine virtuelle. L'entrepôt de données a donc été exporté en csv. Cette modification ne change pas l'aspect conceptuel de cette chaine décisionnelle.

4.1.2 Méthodologie utilisée

Il faut charger tout d'abord les données dans des tables SAS. L'ensemble de ces procédures d'import est disponible en annexe A.3. Lire du csv est facile et l'outil permet une personnalisation assez poussée de ce type d'import.
Vient ensuite la préparation des rapports. La partie Data Mart est selon mois représentée dans l'outil par la création d'une requête destinée au rapport voulu (c'est à dire qui répond à la question posée). Pour chaque rapport il a donc été créé une requête dont le résultat est stocké dans des tables. Ces tables sont pour moi les magasins du shéma proposé en introduction.
Les procédures de création de ces magasins sont en partie données pour exemple en annexe A.4
Les rapports sont ici représentés sous forme de graphiques. L'objet n'étant pas ici de tester tous les types de graphiques possibles, seuls des diagrammes en battons sont ici représentés. En annexe A.5 sont donnés quelques uns des diagrammes proposés pour répondre à certaines questions que l'on pourrait se poser sur ces données.
L'outil ne semble pas proposer de fonction de cube OLAP telle que le drill down ou drill up. Pour l'exercice, des tables ont été créées dans ce sens en jouant sur les différentes hiérarchies.

4.1.3 Conclusions sur l'utilisation de SAS

SAS semble très puissant au niveau statistiques. SAS ne semble pas proposer d'interface ludique permettant de créer facilement ses rapports autrement que par du code SAS. La notion de cube semble lui être aussi nativement inconnue. Si ce n'est pas lié à la version University edition, cela me le ferait exclure de mes choix pour une utilisation en entreprise destinée à des décideurs. Un composant mobile de la version payante semble plus élaboré mais il faudrait pouvoir le tester. Il faudrait aussi voir SAS sur de gros volumes de données. Peut être que l'outil révèle toute sa puissance alors. Il semble pouvoir être configuré pour se servir d'une grille de calculs ce qui le rend intéressant en fonction du besoin. En tout cas, pour un besoin aussi simple que celui que je propose dans ce projet, l'outil ne semble pas forcément adapté.

5 Conclusion

La chaine décisionnelle telle que proposée en introduction est intéressante pour l'exercice mais ne semble pas être la plus optimale pour une utilisation industrialisée en entreprise.
Pour la partie TALEND, j'aimerai tester l'intégration du jar générable dans une application JEE ou tout simplement lancé par un ordonnanceur en batch quotidien. La montée en charge du jar généré doit être intéressante à évaluer aussi. Le fait qu'il s'agisse de JAVA rend possible l'investigation par des experts JAVA notamment à l'aide d'outil de profiling. Même si l'expérience utilisateur a été pour moi désastreuse, je ne l'éliminerai pas d'un choix à faire en entreprise sans évaluation et confrontation au besoin.
L'utilisation de SAS pour la partie reporting ne m'a pas convaincue. Soit je suis passé à côté, soit l'outil semble dédié à un besoin très particulier qui le ferait exclure de nombreuses études de mise en place d'outil de reporting. L'outil semble destiné à des développeurs SAS ayant surtout besoin d'une capacité à traiter d'éléments statistiques sur de grands volumes de données. Chose que je n'ai pas testée. Je ne pense pas qu'il soit destiné directement à des décideurs à part évidemment les rapports produits.

A Annexes

A.1 Code R ayant généré les données

Listing A.1 – generateData.R

```
library(dplyr)
library(tidyr)

setwd('E:/workspace/R/cnam/nfe211')

prenoms <- read.csv('liste_des_prenoms_par_annee.csv', sep = ';')


head(prenoms)
hist(prenoms$nombre)
summary(prenoms$nombre)
hist(x = prenoms$annee)
nb_clients <- length(prenoms$prenoms)


X <- 10 + rnorm(nb_clients, mean=5, sd = 1)
Y <- 5 + rnorm(nb_clients, mean=20, sd = 2)

U <- rbinom(n=nb_clients, size=1, prob=0.6)
summary(U)


hist(X)
```

```
hist(Y)
Z <- U * X + ((U-1)*-1 + Y)
hist(Z)
age <- floor(Z)
clients <- cbind(prenoms, age)
head(clients)
hist(clients$age)


type<-floor(1+rnorm(nb_clients, mean=1.4, sd=0.6)%%3)
hist(type)
clients <- cbind(clients, type)

livres <- c(6, 7, 1, 2, 4, 8, 3, 5)

# distribution sur livres
livres_achetes <- floor(rnorm(nb_clients, mean = 3.5, sd=1.5))
livres_achetes <- sapply(livres_achetes, FUN = function(x) max(0,x))
livres_achetes <- sapply(livres_achetes, FUN = function(x) min(7,x))
hist(livres_achetes)


clients <- cbind(clients, livres_achetes)
head(clients)

# lien vers la table livre
livre <- sapply(livres_achetes, FUN=function(x)livres[x+1])
clients <- cbind(clients, livre)


hist(clients$livre)

boutiques <- floor(runif(nb_clients, min=1, max=25))
clients <- cbind(clients, boutiques)
hist(boutiques)

client_id <- 1:length(clients$prenoms)
```

```
clients <- cbind(clients, client_id)

#last date in the data
end_date <- as.Date('2015-01-01')

date_achat <- end_date - runif(n=nb_clients, max=365*5, min=1)



clients <- cbind(clients, date_achat)
clients <- mutate(clients, prenom = prenoms)

clients_for_csv <- select(clients, id=client_id, age, prenom)
write.csv(x=clients_for_csv, file='clients.csv', row.names=FALSE)
nb_clients + 1 - client_id

clients <- mutate(clients, facture_id = (nb_clients + 1 - client_id))

facture_for_csv <- select(clients, id=facture_id, date_achat, boutique_id=b
write.csv(x=facture_for_csv, file='factures.csv', row.names=FALSE)

villes <- read.csv(file='ville.csv')
departements <- read.csv('departement.csv')
regions <- read.csv('region.csv')
```

A.2 Routine java toTrancheDage.java

Listing A.2 – toTrancheDage.java

```
public class to_tranche_age {
        private enum TRANCHE_AGE{
                LT_12("-12"),
                B12_25("12-25"),
                B25_34("25-34"),
                B35_65("35-65"),
                MT_65("+65");

                TRANCHE_AGE(String label){
                        this.label = label;
                }

                private final String label;

        }

    public static String perform(int age){
        if(age < 0){
                throw new IllegalArgumentException(
                        "age should not be less than 0: " + age);
        }
        if(age < 12){return TRANCHE_AGE.LT_12.label;}
        if(age >= 12 && age <= 25){return TRANCHE_AGE.B12_25.label;}
        if(age > 25 && age <= 34){return TRANCHE_AGE.B25_34.label;}
        if(age > 34 && age <= 65){return TRANCHE_AGE.B35_65.label;}
        return TRANCHE_AGE.MT_65.label;
    }
}
```

A.3 import.sas

Listing A.3 – import.sas

```
data WORK.SAS_CLIENT      ;
```

```
%let _EFIERR_ = 0; /* set the ERROR detection macro variable */
infile '/folders/myfolders/Talend_client.csv'
        delimiter = ','
        MISSOVER DSD lrecl=32767
        firstobs=2 encoding='wlatin1';

        informat id best32. ;
        informat prenom $9. ;
        informat age best32. ;
        informat tranche_age $5. ;
        format id best32. ;
        format prenom $9. ;
        format age best32. ;
        format tranche_age $5. ;
input
        id
        prenom $
        age
        tranche_age $
        ;
        if _ERROR_ then call symputx('_EFIERR_',1);
/* set ERROR detection macro variable */
run;

PROC IMPORT
DATAFILE="/folders/myfolders/Talend_vente.csv"
out=sas_vente
DBMS=CSV

replace
;
GETNAMES=YES;

run;

PROC IMPORT
DATAFILE="/folders/myfolders/Talend_temps.csv"
```

```
out=sas_temps
DBMS=CSV

replace
;
GETNAMES=YES;

run;

PROC IMPORT
DATAFILE="/folders/myfolders/Talend_livre.csv"
out=sas_livre
DBMS=CSV

replace
;
GETNAMES=YES;

run;

data WORK.SAS_BOUTIQUE    ;
 %let _EFIERR_ = 0; /* set the ERROR detection macro variable */
 infile '/folders/myfolders/Talend_boutique.csv'
        delimiter = ','
        MISSOVER DSD lrecl=32767 firstobs=2 encoding='wlatin1';
 informat id best32. ;
 informat nom $29. ;
 informat adresse $28. ;
 informat ville $13. ;
 informat departement $14. ;
 informat region $15. ;
 format id best12. ;
 format nom $29. ;
 format adresse $28. ;
 format ville $13. ;
 format departement $14. ;
 format region $15. ;
```

```
input
id
nom $
adresse $
ville $
departement $
region $
;
if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro
run;
```

A.4 reporting.sas

Listing A.4 – reporting.sas

```
/*Quelle age achete le plus de livre*/
proc sql;
create table vente_client as
select sv.id, age, tranche_age from work.SAS_CLIENT SC
full join WORK.SAS_VENTE SV
on SC.id = SV.client_id;
quit; run;

title "Quelle age achete le plus de livre";
proc sgplot data=vente_client;
        histogram age / transparency=0.7 binwidth=1;
        yaxis grid;
        xaxis display=(nolabel);
run;

/*Quelle tranche d'age achete le plus de livre*/
title "Quelle tranche d'age achete le plus de livre";
proc sgplot data=vente_client;
        vbar tranche_age / transparency=0.5 ;
run;
```

```
/*Dans quelle ville fait on le plus gros
chiffre d'affaire de livre en 2014*/
proc sql;
create table vente_boutique_temps_2014_somme as
select bout.ville, sum(sv.prix) as CA_VILLE from work.sas_vente sv
full join work.sas_boutique bout
on bout.id = sv.boutique_id
full join work.sas_temps tmp
on tmp.id = sv.temps_id
where tmp.annee = 2014
group by ville;

quit;run;

/*Dans quelle ville fait on le plus gros
chiffre d'affaire de livre en 2014*/
title "Dans quelle ville fait on le plus
gros chiffre d'affaire de livre en 2014";
proc sgplot data=vente_boutique_temps_2014_somme;
        vbar ville / transparency=0.6 response=CA_VILLE;
run;


/*Dans quelle region fait on le plus gros
 chiffre d'affaire de livre en 2014*/
proc sql;
create table vente_boutique_2014_somme_region as
select bout.region, sum(sv.prix) as CA_REGION
from work.sas_vente sv, work.sas_boutique bout, work.sas_temps tmp
where tmp.annee = 2014
and bout.id = sv.boutique_id
and tmp.id = sv.temps_id

group by region;

quit;run;
```

```
/*Dans quelle region fait on le plus gros
chiffre d'affaire de livre en 2014*/
title "Dans quelle region fait on le plus gros
chiffre d'affaire de livre en 2014";
proc sgplot data=vente_boutique_2014_somme_region;
        vbar region / transparency=0.6 response=CA_REGION;
run;

proc print data=vente_boutique_2014_somme_region;
        var region CA_REGION;
        title "CA des regions";
run;

/*Dans quelle ville vend-on le plus de livre en 2014*/
proc sql;
create table vente_boutique_temps_2014 as
select sv.id, tmp.annee, tmp.mois, bout.ville, bout.region
from work.sas_vente sv
full join work.sas_boutique bout
on bout.id = sv.boutique_id
full join work.sas_temps tmp
on tmp.id = sv.temps_id
where tmp.annee = 2014;
quit;run;


/*Dans quelle ville vend-on le plus en 2014*/
title "Dans quelle ville vend-on le plus en 2014";
proc sgplot data=vente_boutique_temps_2014;
        vbar ville / transparency=0.6 ;
run;

/*Dans quelle region vend-on le plus en 2014*/
title "Dans quelle region vend-on le plus en 2014";
proc sgplot data=vente_boutique_temps_2014;
        vbar region / transparency=0.6 ;
```

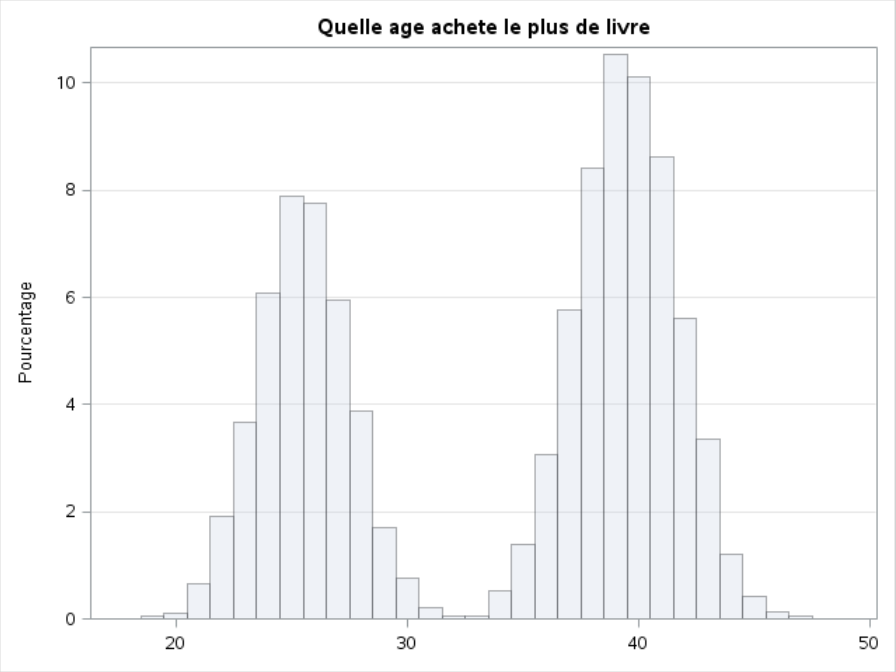
```
run;

title "nombre de ville par region";
proc sgplot data=SAS_boutique;
        vbar region / transparency=0.6 ;
run;

/*Comment sont reparties les ventes
sur l'annee en region bretagne*/
proc sql;
create table vente_boutique_temps_bretagne as
select sv.id, tmp.mois from work.sas_vente sv
full join work.sas_boutique bout
on bout.id = sv.boutique_id
full join work.sas_temps tmp
on tmp.id = sv.temps_id
where bout.region = 'Bretagne';
quit;run;

/*Comment sont reparties les ventes
sur l'annee en region bretagne*/
title "Comment sont reparties les ventes
sur l'annee en region bretagne";
proc sgplot data=vente_boutique_temps_bretagne ;
        vbar mois / transparency=0.6 ;
run;

/*Comment evoluent les ventes dans le temps*/
proc sql;
create table vente_temps as
select sv.id, tmp.annee from work.sas_vente sv
full join work.sas_temps tmp
on tmp.id = sv.temps_id;
quit;run;

/*Comment evoluent les ventes dans le temps*/
title "Comment evoluent les ventes dans le temps";
```

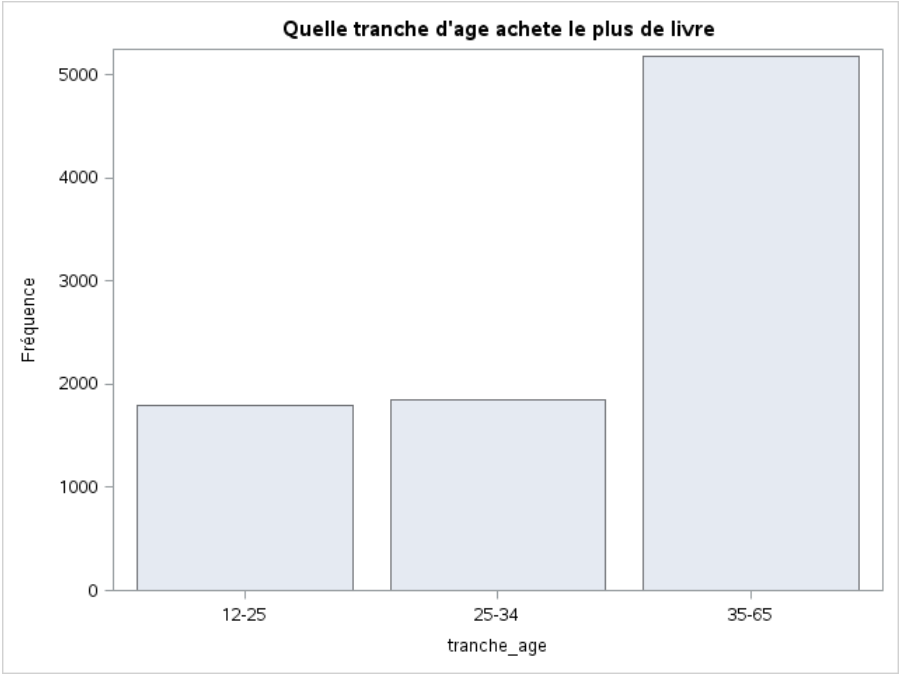
```
proc sgplot data=vente_temps ;
        vbar annee/ transparency=0.6 ;
run;

/*Comment sont repartis les achats de livres*/
proc sql;
create table vente_livre as
select sas_vente.id, nom, genre
from sas_vente
inner join sas_livre
on sas_livre.id = sas_vente.livre_id;
quit;run;

/*Comment sont repartis les achats de livres*/
title "Comment sont repartis les achats de livres";
proc sgplot data=vente_livre;
        vbar nom / transparency=0.6;
run;

/*Comment sont repartis les achats de livres*/
proc sql;
create table vente_livre_somme as
select nom, sum(prix) as CA_LIVRE
from sas_vente
inner join sas_livre
on sas_livre.id = sas_vente.livre_id
group by nom;
quit;run;

/*Comment sont repartis les achats de livres*/
title "Comment sont repartis les achats de livres";
proc sgplot data=vente_livre_somme;
        vbar nom / transparency=0.2 response=CA_LIVRE;
run;



/*Comment sont repartis les achats de livres*/
```

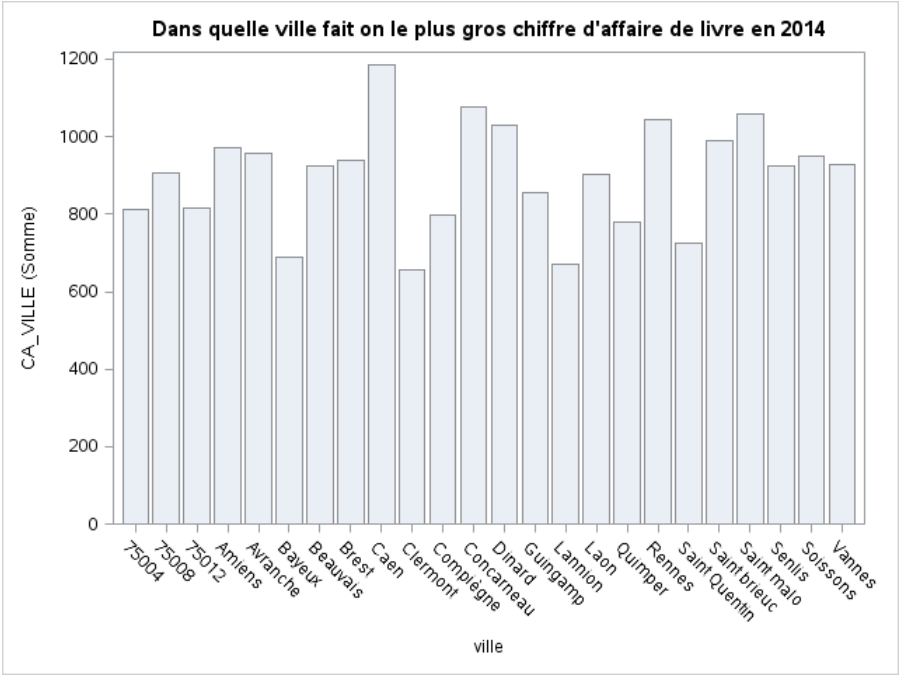
```
title "Comment sont repartis les achats de livres";
proc sgplot data=vente_livre;
        vbar genre/ transparency=0.6;
run;

/*Comment sont repartis les achats de livres par genre*/
proc sql;
create table vente_livre_somme_genre as
select genre, sum(prix) as CA_LIVRE
from sas_vente
inner join sas_livre
on sas_livre.id = sas_vente.livre_id
group by genre;
quit;run;

/*Comment sont repartis les achats de livres par genre*/
title "Comment sont repartis les achats de livres par genre";
proc sgplot data=vente_livre_somme_genre;
        vbar genre / transparency=0.2 response=CA_LIVRE;
run;

/*Comment sont repartis les achats
de livres par genre et par annee*/
proc sql;
create table vente_livre_annee_genre as
select annee, genre, sum(prix) as CA_LIVRE
from sas_vente
inner join sas_livre
on sas_livre.id = sas_vente.livre_id
inner join sas_temps
on sas_vente.temps_id = sas_temps.id
group by annee, genre;
quit;run;

/*Comment sont repartis les achats
de livres par genre et par annee*/
title "Comment sont repartis les achats
```

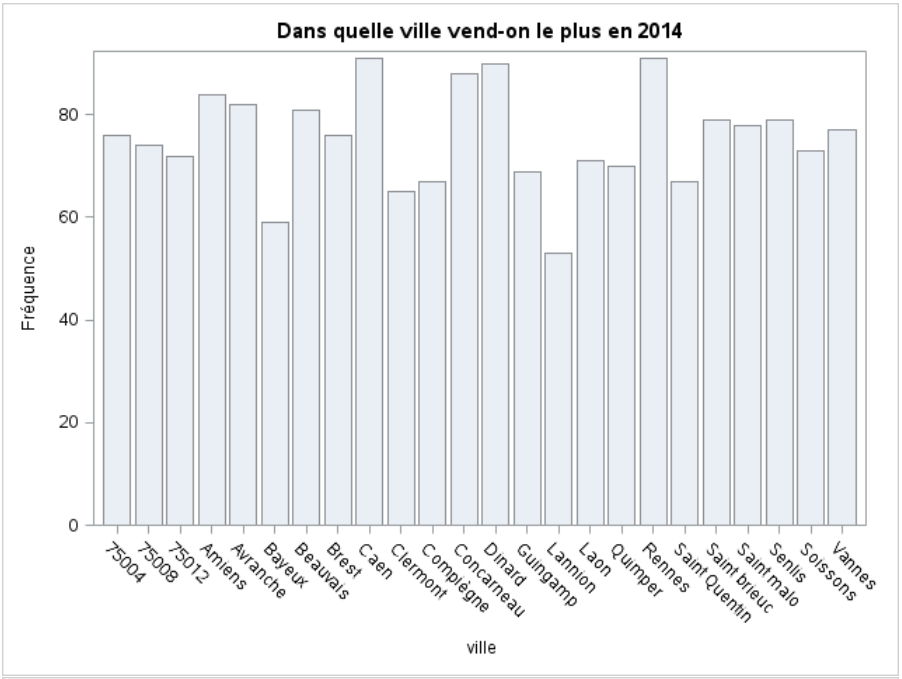
```
de livres par genre et par annee";
proc sgplot data=vente_livre_annee_genre ;
        vbar annee
        / transparency=0.2
        group=genre groupdisplay=cluster response=CA_LIVRE;
run;
```

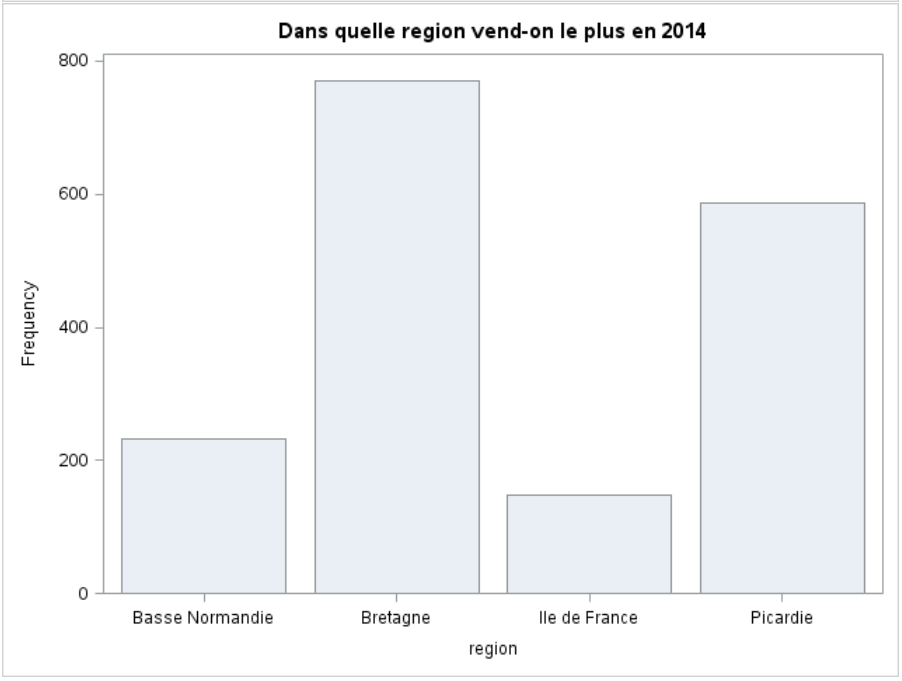
A.5 report sas

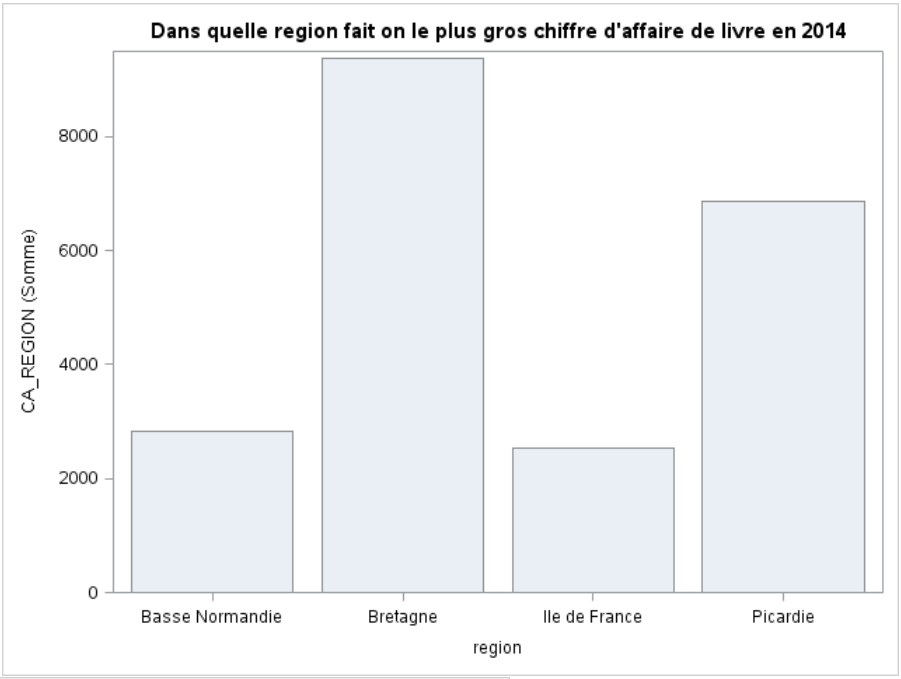



Quelle tranche d'age achete le plus de livre
Fréquence
5000
4000
3000
2000
1000
0
12-25
25-34
35-65
tranche_age


Dans quelle ville fait on le plus gros chiffre d'affaire de livre en 2014
CA_VILLE (Somme)
1200
1000
800
600
400
200
0
75004
75008
75012
Amiens
Avranche
Bayeux
Beauvais
Brest
Caen
Clermont
Compiègne
Concarneau
Dinard
Guingamp
Lannion
Laon
Quimper
Rennes
Saint Quentin
Saint brieuc
Saint malo
Senlis
Soissons
Vannes
ville

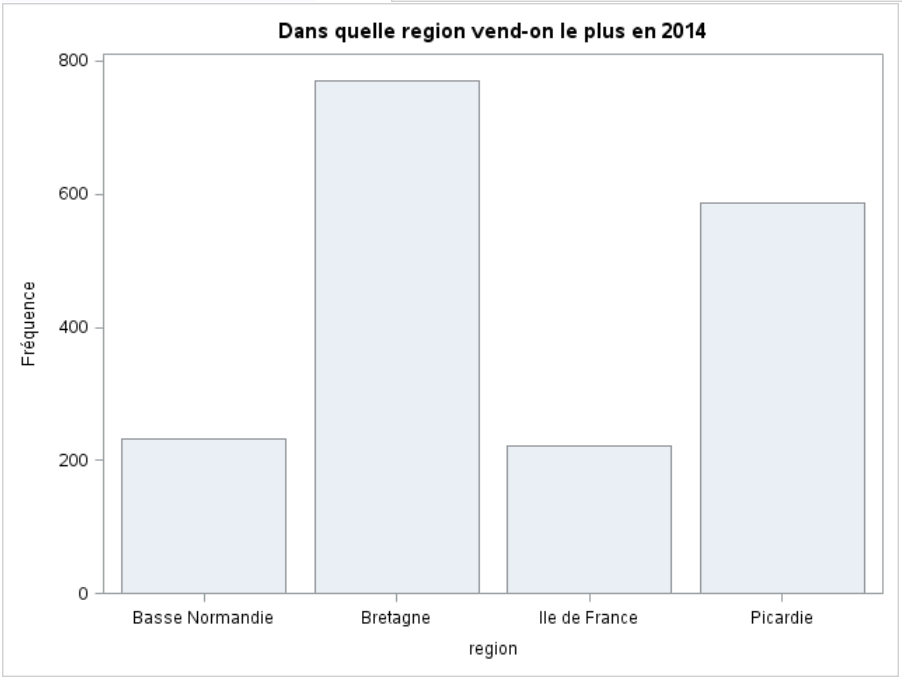

Dans quelle ville vend-on le plus en 2014
Fréquence
0
20
40
60
80
75004
75008
75012
Amiens
Avranche
Bayeux
Beauvais
Brest
Caen
Clermont
Compiègne
Concarneau
Dinard
Guingamp
Lannion
Laon
Quimper
Rennes
Saint Quentin
Saint brieuc
Saint malo
Senlis
Soissons
Vannes
ville

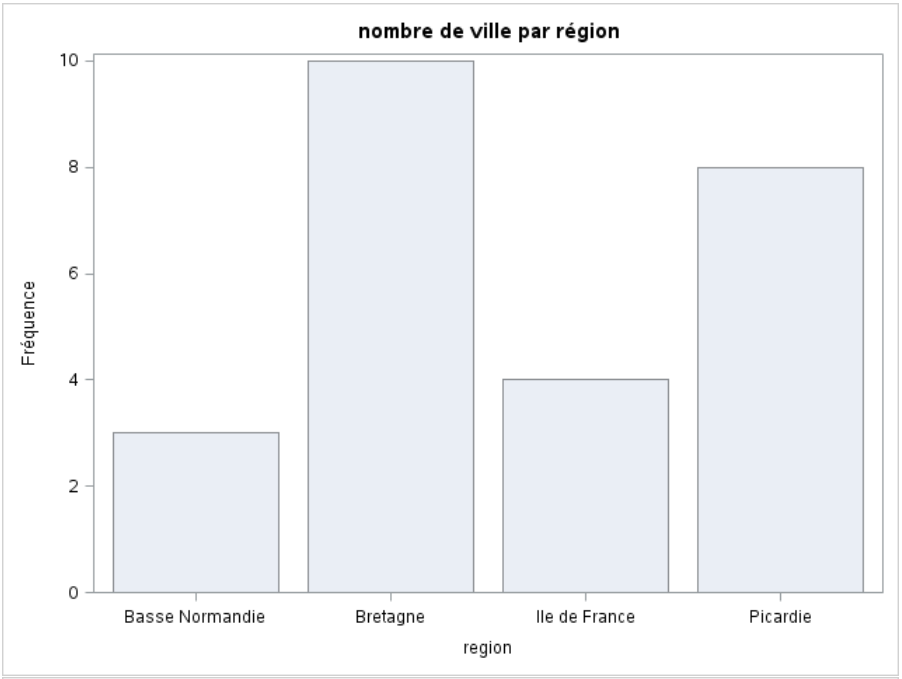

Dans quelle region vend-on le plus en 2014
Frequency
0
200
400
600
800
Basse Normandie
Bretagne
Ile de France
Picardie
region

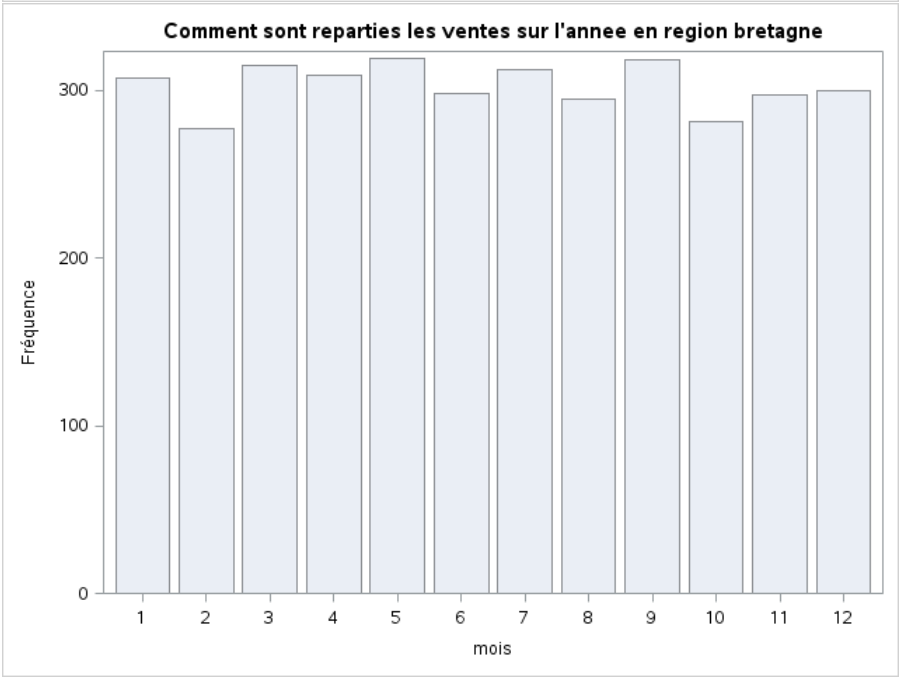


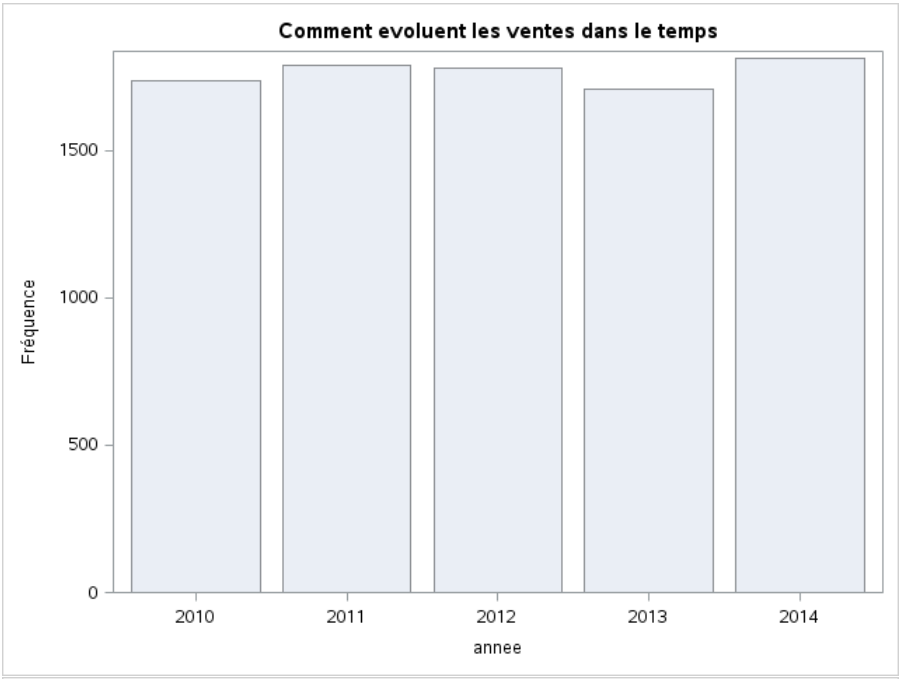
CA des regions

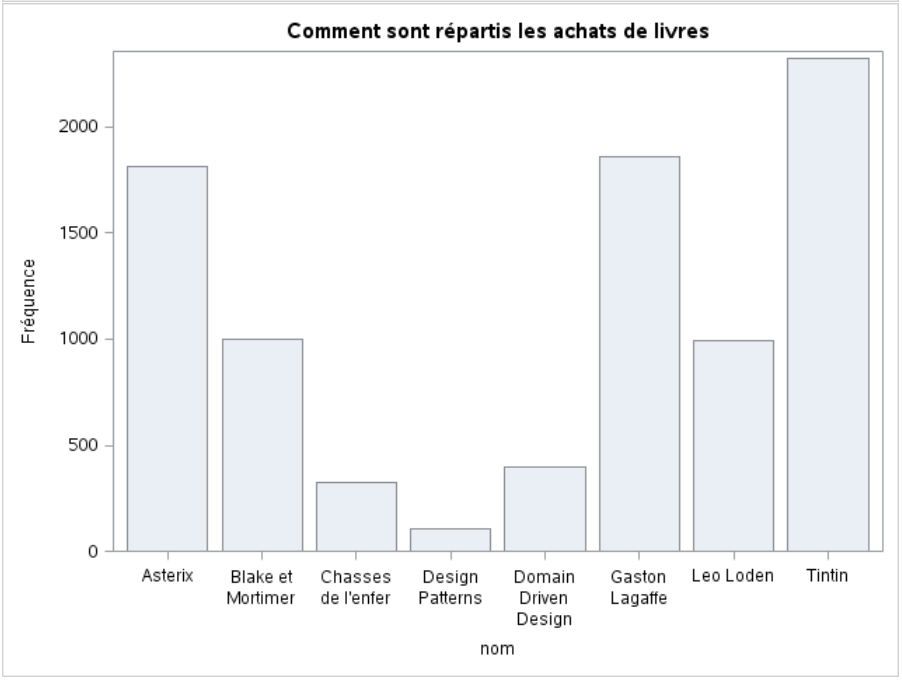
Obs.	region	CA_REGION
1	Basse Normandie	2835.00
2	Bretagne	9369.25
3	Ile de France	2534.80
4	Picardie	6849.55

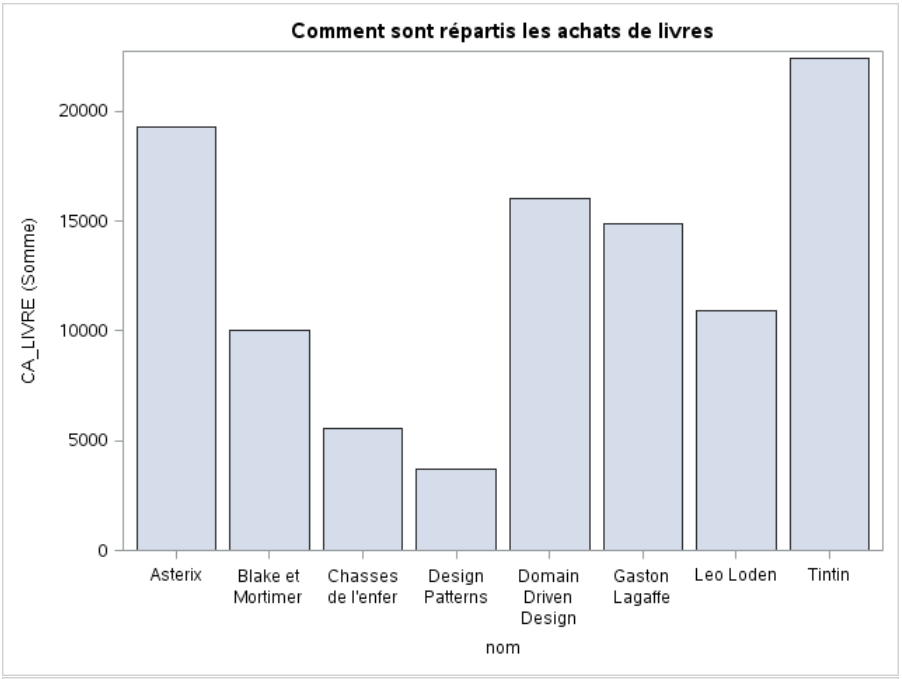


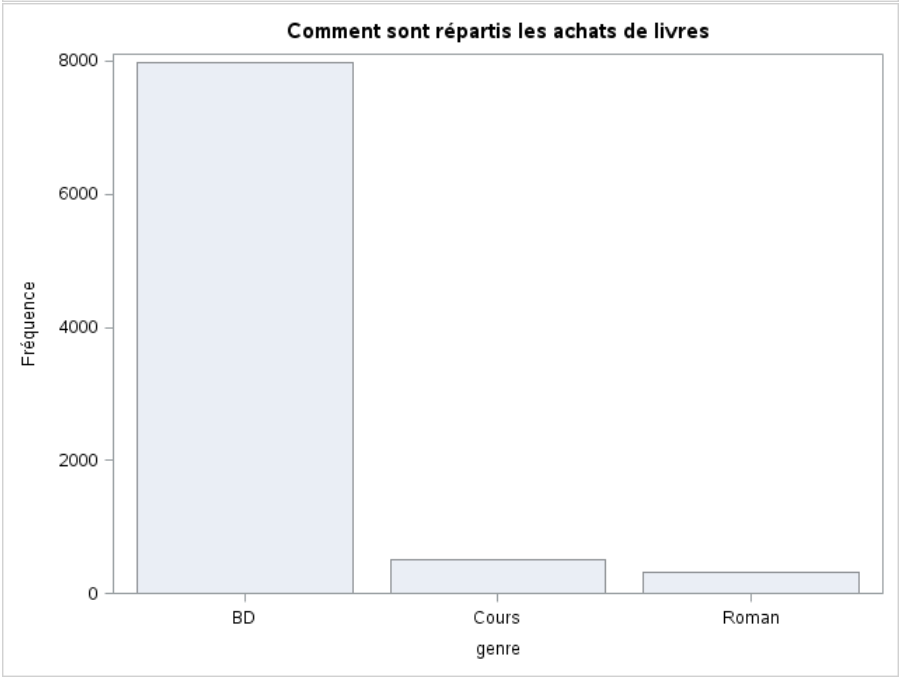

nombre de ville par région
Fréquence
10
8
6
4
2
0
Basse Normandie
Bretagne
Ile de France
Picardie
region

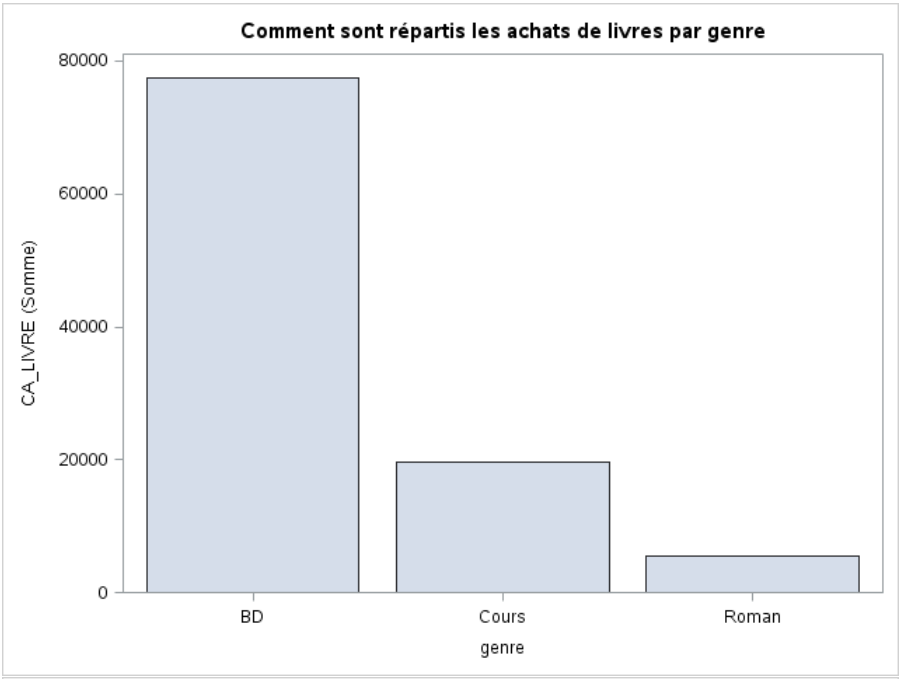

Comment sont reparties les ventes sur l'annee en region bretagne
Fréquence
300
200
100
0
1
2
3
4
5
6
7
8
9
10
11
12
mois

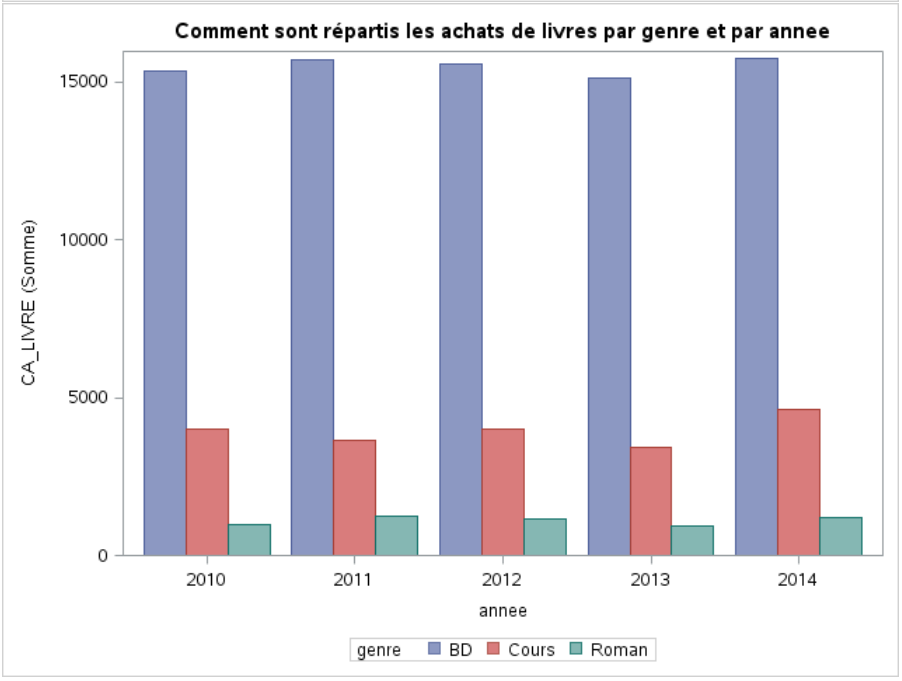

Comment evoluent les ventes dans le temps
Fréquence
1500
1000
500
0
2010
2011
2012
2013
2014
annee


Comment sont répartis les achats de livres
Fréquence
2000
1500
1000
500
0
Asterix
Blake et Mortimer
Chasses de l'enfer
Design Patterns
Domain Driven Design
Gaston Lagaffe
Leo Loden
Tintin
nom


Comment sont répartis les achats de livres
CA_LIVRE (Somme)
20000
15000
10000
5000
0
Asterix
Blake et Mortimer
Chasses de l'enfer
Design Patterns
Domain Driven Design
Gaston Lagaffe
Leo Loden
Tintin
nom


Comment sont répartis les achats de livres
Fréquence
8000
6000
4000
2000
0
BD
Cours
Roman
genre


Comment sont répartis les achats de livres par genre
80000
60000
40000
20000
0
CA_LIVRE (Somme)
BD
Cours
Roman
genre


Comment sont répartis les achats de livres par genre et par annee
15000
10000
5000
0
CA_LIVRE (Somme)
2010
2011
2012
2013
2014
annee
genre
BD
Cours
Roman