Analyse des données

L'analyse des données a pour but de recomposer l'information contenue dans les données recueillies afin d'en fournir une vue plus synthétique, ou d'extraire des informations qui n'apparaissent pas de façon explicite dans la série de mesures initiale.

* **Analyse des données**

- faire ressortir des corrélations (exemple : type d'habitat/intentions de vote),
- pour des enquêtes de consommation, du marketing ciblé ...

Le but est de dégager :

- des **tendances** (covariables)
- des **modes** de la distribution (présence de plusieurs maxima)

à partir d'un **grand** ensemble de données (chiffre d'affaires, nb de ventes, masse salariale, ...) évoluant dans le **temps** et dans l'**espace**, afin de

- définir des **indicateurs** pertinents
- faciliter la prise de décision.

Exemples de grandes masses de données :

- Masses de données (pullulantes) : tickets de caisse, clics web, appels tel, operations bancaires, remboursements no URSSAF, trajets SNCF...
- Données importantes : fichiers de clients, données biométriques, campagnes de mesures, sondages,...
- Données géographiquement localisées (gestion d'un "territoire") : appels tel, centres de production, consommation eau-électricité-gaz, infractions pénales, arrêts maladie, prêts bancaires, allocations chômage, jugements des TGI, accidents du travail...

Sites de données :

- www.data.gouv.fr
- www.datasud.fr
- scientific data
- Liste de ressources open data
- data.gov
- open food facts

Principales méthodes d'analyse

- **REPRESENTATION DES DONNEES** : Représenter des jeux de valeurs de grande taille de façon plus synthétique (algorithmes de réduction de dimension)
- **REGROUPEMENT ("CLUSTERING")** : Définir des regroupements (ou des classements simples ou hiérarchiques) entre jeux de valeurs
- **COMPLETION** : Méthodes de classification automatique (ou d'interpolation) visant à deviner soit la classe, soit certaines valeurs non mesurées, à partir d'un jeu de valeurs et d'une base d'exemples complets. Il existe des méthodes

paramétriques ou non paramétriques.

- **ESTIMATION ET DECISION** : Méthodes visant à estimer la “valeur” associée à un jeu de données (pour l’aide à la décision)

2. L'agrégation

Rappel

On distingue classiquement deux grandes catégories de données :

- données **quantitatives**:
  - numérique entier ou réel, discrètes ou continues, bornées ou non. ex: poids, taille, âge, taux d’alcoolémie,...
  - temporel : date, heure
  - numéraire
  - etc...
- données **qualitatives**:
  - de type vrai/faux (données booléennes). ex: marié/non marié, majeur/non majeur
  - de type appartenance à une classe. ex: célibataire/marié/divorcé/veuf, salarié/chômeur/retraité etc...
  - de type texte (autrement dit “chaîne de caractères”). ex: nom, prénom, ville,...

Les données qualitatives:

- définissent l'appartenance à une catégorie
- permettent de définir des **classes** au sein d'un ensemble de données

Organisation des données sous forme de tableaux bidimensionnels

Schémas de données

- Un enregistrement est un jeu de valeurs organisé sous forme de **tuple**
- A un tuple on associe en général un **schéma de données**.

	Nom	Prénom	Adresse	Âge
SCHEMA :				
DONNEES :	Dubois	Martine	29, rue du Verger, Orléans	22

- Définir un **schéma** consiste à définir :

    - une liste d'attributs (labels) associées à chacune des valeurs du tuples.
- A chaque **attribut** correspond :
    - un *intitulé*
    - un *domaine* de valeurs (type/format des données)

Tableau de données

Un tableau de données est une liste (finie et ordonnée) de tuples, chaque tuple obéissant à un même schéma R.

Tableau de données

Nom	Prénom	Adresse	Âge
Dubois	Martine	29, rue du Verger, Orléans	22
Gilbert	Jonas	8, rue des Fleurs, Blois	23
Dalban	Pierre	13, av. du Général, Privas	22
...	...	...	...
Manoukian	Marianne	55, place des Bleuets, Aubagne	24

schéma

tuple

L'indexation

- D’un point de vue informatique, le jeu de valeurs recueilli est appelé un *enregistrement*, correspondant à l’encodage des données recueillies dans un format numérique
- Pour une gestion efficace des données, il est nécessaire de pouvoir identifier chaque enregistrement de façon unique.

Indexation simple

Définition

- L'indexation des données consiste à attribuer à chaque donnée distincte un identifiant unique.
- On parle également de *clé* du jeu de données:
- On peut représenter l'opération d'indexation sous la forme d'une fonction:
    - Si d est le jeu de valeurs
    - $id(d)$ désigne l'identifiant de ce jeu de valeurs

- L'indexation des données :
    - repose sur un principe simple d'étiquetage

- consistant à attribuer une étiquette différente à chaque enregistrement.
- Cette étiquette peut être
  - une suite de caractères arbitraires,
  - un entier,
  - ou un descripteur explicite.

Exemples d'indexation centralisée :

- numéro INE (étudiants)
- numéro URSSAF (sécurité sociale)
- numéro d'immatriculation (véhicules)
- numéro de compte en banque
- code barre
- etc.

La recherche par index repose sur une **fonction d'adressage** ref qui à tout identifiant id associe la référence de la donnée correspondante: $$\text{ref} : \text{id} \rightarrow d = \text{ref}(\text{id}) $$ où d est l'"adresse" du jeu de données.

Exemples d'adressages:

- Table des matières (en-têtes de chapitres) –> num de page
- nom de variable –> valeur
- chemin d'accès (fichiers) –> adresse des données sur le support physique
- URL –> adresse IP (L'index est fourni par un serveur de noms)
- code d'accès –> contenu

L'exemple le plus simple d'indexation est celui fourni par les **numéros de case** d'un tableau.

- Soit D un tableau de n lignes
- le numéro $i < n$ peut être vu à la fois l'identifiant et l' *adresse* de la ligne $D[i]$

Index	Données			
0	Dubois	Martine	29, rue du Verger, Orléans	22
1	Gilbert	Jonas	8, rue des Fleurs, Blois	23
2	Dalban	Pierre	13, av. du Général, Privas	22
	...	...	...	...
n - 1	Manoukian	Marianne	55, place des Bleuets, Aubagne	24

Mise en oeuvre :

Définition

On appelle index la structure de données qui implémente la fonction d'adressage

- Listes :
  - La lecture de l'index repose sur le parcours d'une liste

$$I = ((\text{id}_1, d_1), (\text{id}_2, d_2), ..., (\text{id}_n, d_n))$$ telle que $\text{id}_1 < \text{id}_2 < ... < \text{id}_N$, de telle sorte que la recherche s'effectue en O(log n) (recherche dichotomique).

- Dictionnaires :

$$ I = \{\text{id}_1:d_1, \text{id}_2:d_2, ..., \text{id}_n:d_n\}$$

- La structure de dictionnaire permet une recherche en $O(1)$.

Index de partitionnement

Un index de partitionnement:

- est constitué d'un ensemble fini de classes $\mathcal{K}$
- à chaque classe sont associées plusieurs références de la table de données $\mathcal{D}$
- autrement dit :

$$ \forall \text{c} \in \mathcal{K}, \text{c} \rightarrow D_c = \{d_1, d_2, ...\} \subset \mathcal{D}$$
$$ \forall d \in \mathcal{D}, \exists! \text{c} \text{ t.q. } d \rightarrow \text{c}$$

- On dit que les données sont organisées en *classes* : chaque donnée appartient à une classe unique c
- L'ensemble de classes $\mathcal{K}$ définit une partition du tableau de données $\mathcal{D}$.

Implémentations:

- Dictionnaire de listes :

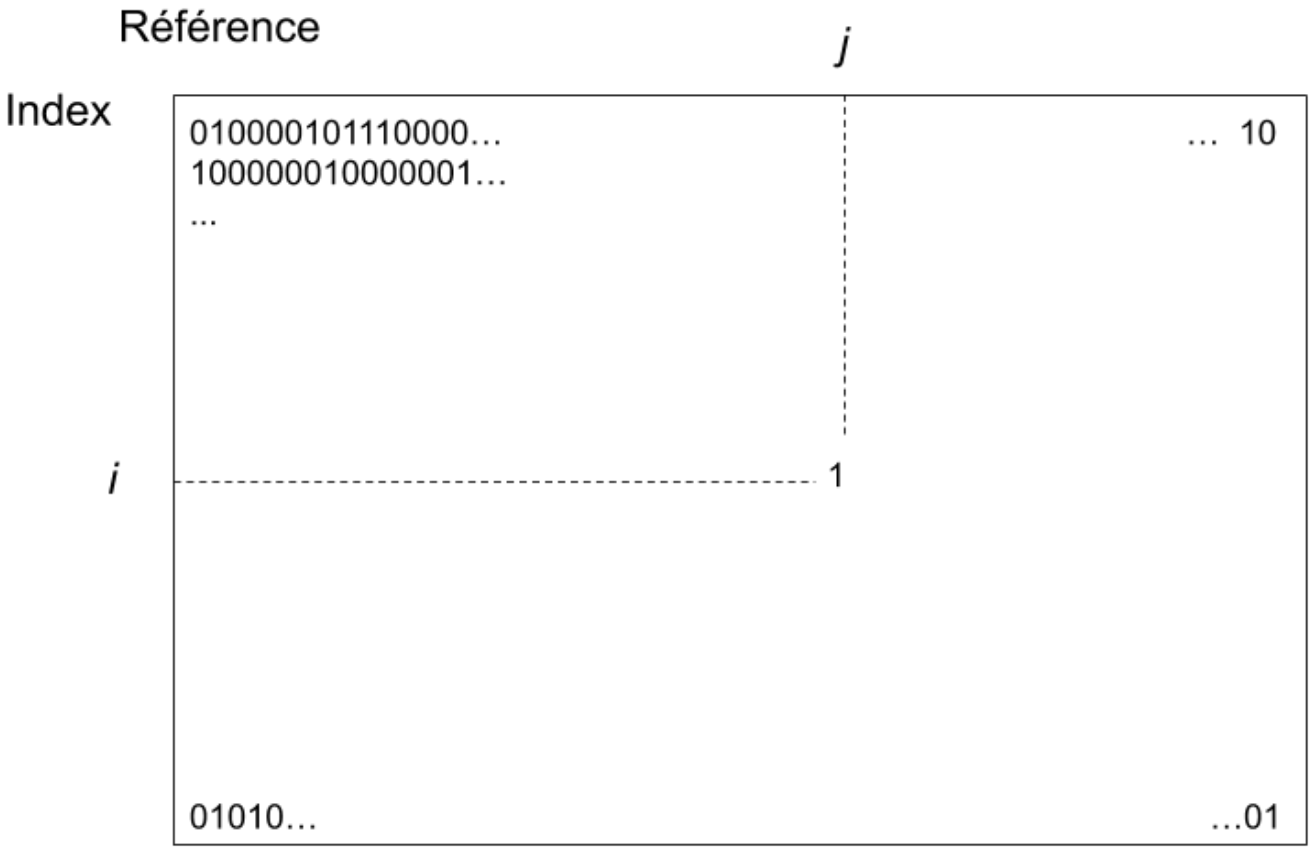
$$ I = \{c_1:(d_{11}, d_{12}, ...), c_2:(d_{21}, d_{22}, ...), ... \}$$

- à chaque classe est associée une *liste* de références

- Index "Bitmap":
  - Soit $\mathcal{D}$ un ensemble de n enregistrements

- Soit $\mathcal{K}$ un index qui partitionne $\mathcal{D}$
- avec $m = |\mathcal{K}|$: taille de l'index de partitionnement
- un entier j dans {1,...,n} est attribué à chaque enregistrement
- un entier i dans {1,...,m} est attribué à chaque classe
- L'index est défini comme une matrice creuse M avec:

$$ M[i,j] = 0 \Leftrightarrow d_j \notin D_i $$ $$ M[i,j] = 1 \Leftrightarrow d_j \in D_i $$



Agrégation

L'agrégation consiste à partitionner les données en classes selon la **valeur** d'un attribut.

Il est supposé que le partitionnement s'effectue sur des attributs pour lequel le nombre de valeurs possibles est fini.

Si A est l'attribut considéré:

- A est un attribut "qualitatif"
- Le domaine de valeurs de A $\text{dom}(A)$ est fini
- $\text{dom}(A)$ définit un index de partitionnement

Une fois la partition effectuée, il est courant d'effectuer des **mesures** sur chaque partition obtenue

Les mesures sont réalisées à l'aide d'**opérateur d'agrégation** :

- comptage, somme, moyenne, ecart-type, max, min, ...
- (`count`, `sum`, `mean`, `avg`, `max`, `min`...)

cas d'utilisation :

- Quels sont les catégories de films/livres les plus fréquemment empruntés?
- A quelles heures de la journée la messagerie est-elle la plus sollicitée?
- Comment se répartissent géographiquement les utilisateurs de la messagerie?

SQL

En SQL, l'opérateur d'agrégation est `GROUP BY`:

- partitionne les données à partir des valeurs de l'attribut mentionné
- il est possible de partitionner les données selon les valeurs de plusieurs attributs

Exemples de requêtes faisant appel aux fonctions d'agrégation :

Nombre d'élèves par groupe de TD / par prépa d'origine etc..:

```
SELECT groupe_TD , COUNT(num_eleve)
FROM Eleve
GROUP BY groupe_TD
```

Donner les chiffres des ventes du magasin pour chaque mois de l'année

```
SELECT mois, SUM(montant)
FROM Vente
GROUP BY mois
```

Donner le nombre de ventes d'un montant > à 1000 euros pour les mois dont le chiffre d'affaires est supérieur à 10000 euros

```
SELECT mois, COUNT(num_vente)
FROM Vente
GROUP BY mois
HAVING SUM(montant) >= 10000
```

Tester les disparités salariales entre hommes et femmes

```
SELECT sexe, avg( salaire )
FROM Employé
GROUP BY sexe
```

Tester les disparités salariales selon le niveau d'éducation

```
SELECT niveau_educatif, avg( salaire )
FROM Employé
```

```
GROUP BY niveau_éducatif
```

Pandas

L'utilisation de données structurées dans un programme Python nécessite de faire appel à des librairies spécialisées. Nous regardons ici la librairie `pandas` qui sert à la mise en forme et à l'analyse des données.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas
```

Considérons des informations stockées dans un fichier au format 'csv' (comma separated values) : ventes_new.csv

On utilise:

- `pandas.read_csv`. Voir dataframes pandas. Pandas permet également de lire les données au format `xls` et `xlsx` (Excel).

```
with open('ventes_new.csv') as f:
    data = pandas.read_csv(f)
print(data)
```

avec `data` une structure de données de type `DataFrame`.

Pandas offre la possibilité d'organiser et analyser les données par *classe*.

Le partitionnement repose sur des valeurs d'attributs (il y a autant de groupes qu'il y a de valeurs différentes pour l'attribut considéré)

Par exemple si on prend le type de produit:

```
partition_selon_produit = data.groupby('TYPE_PRODUIT')
```

ici l'objet `partition_selon_produit` est une partition du tableau de données selon la valeur de `TYPE_PRODUIT`.

Pour visualiser les partitions:

```
print(partition_selon_produit.groups)
```

On peut ensuite effectuer des mesures et calculs par groupes. Par exemple :

```
nb_ventes_par_produit = partition_selon_produit.size()
```

l'objet `nb_ventes_par_produit` est une série indexée par les valeurs d'attributs (ici 'Bateaux', 'Avions' etc...)

```
print(nb_ventes_par_produit.index)
```

On peut bien sûr l'afficher :

```
print(nb_ventes_par_produit)
```

Les fonctions sum(), mean(), max(), min() etc... s'appliquent sur des mesures, ici MONTANT ou QUANTITE.

Exemple : le chiffre d'affaires par produit (somme des montants) :

```
CA_par_produit = partition_selon_produit.MONTANT.sum()
```

Enfin on peut également effectuer une sélection sur les valeurs calculées (l'équivalent du HAVING en SQL).

Exemples:

- les produits générant un chiffre d'affaires > 1000000:

```
print(CA_par_produit[CA_par_produit > 1000000])
```

- le produit générant le plus haut chiffre d'affaires:

```
print(CA_par_produit[CA_par_produit == max(CA_par_produit)])
```

Les groupes peuvent être définis sur des critères multiples :

```
partition_pays_ville = data.groupby(['PAYS', 'VILLE'])
```

Affichage et figures

```
plt.figure()
nb_ventes_par_produit.plot(kind = "bar", figsize = (5,3))

plt.figure()
nb_ventes_par_produit.plot(kind = "pie", figsize = (5,3))
```

Pour aller plus loin :

- Une introduction très détaillée aux DataFrames (en Français)
- Introduction to Pandas (en anglais)

Tables pivot

- Notion de fait élémentaire (*fact*): transaction ou opération localisée dans le temps et dans l'espace

Remarque : Les transactions marchandes sont un cas classique (acte d'achat bien répertorié et enregistrés, livres de comptes, ...)

Exemples de "fait":

- Achat/Vente
- Opération bancaire (débit/crédit)
- Consultation (site web)
- Souscription à un contrat d'assurance
- Appel téléphonique
- Inscription

Tous ces faits peuvent être localisés. Des mesures peuvent être effectuées sur ces faits (montant d'une vente, durée d'un appel, montant d'une opération bancaire, ...)

Points clés :

- distinction entre **Dimension** et **Mesure**.
  - Notion de dimension : qui? quoi? où? quand? Comment? : associe des **coordonnées** à l'événement (géographiques, temporelles) et par extension une catégorie.
  - Notion de **mesure(s)** associées à l'événement (exemple : montant de la vente)
- distributions, répartitions, etc... cf analogie proba/stats : événement aléatoire, vecteur aléatoire, ...
  - les événements sont associés par paquets sur des intervalles réguliers ou selon des catégories discretes.
  - Fonctions d'aggrégation : réalise la mesure sur les groupe : somme, comptage, moyenne, min, max, etc...
  - histogramme : nb d'événements observés par secteur sur un maillage régulier de l'espace des coordonnées. Par extension mesure sur ce maillage par une fonction d'aggrégation.

principe :

- les mesures portent sur des données de type *quantitatif*
- les classes reposent sur des données de *type qualitatif*.

Les tables pivot permettent d'analyser des faits selon deux dimensions organisées sur les deux axes d'un tableau.

L'utilisation de données structurées dans un programme Python nécessite de faire appel à des librairies spécialisées. Nous regardons ici la librairie `pandas` qui sert à la mise en forme et à l'analyse des données.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas
```

Considérons des informations stockées dans un fichier au format 'csv' (comma separated values) : `ventes_new.csv`

On utilise:

- `pandas.read_csv`. Voir dataframes pandas. Pandas permet également de lire les données au format `xls` et `xlsx` (Excel).

```
with open('ventes_new.csv') as f:
    data = pandas.read_csv(f)
print(data)
```

avec `data` une structure de données de type `DataFrame`.

exemple : on représente les ventes selon (1) la dimension géographique et (2) la dimension temporelle

```
T = pandas.pivot_table(data, values = 'MONTANT', index = ['PAYS'], columns = ['ANNEE'], aggfunc=np.sum)
print(T)
```

Evolution des ventes au cours de l'année pour la France seulement:

```
selection = data[data.PAYS == "France"]
T2 = pandas.pivot_table(selection, values = 'MONTANT', index = ['ANNEE'], columns = ['VILLE'],  aggfunc=np.sum)
print(T2)

T2.plot(kind='bar', subplots = 'True')
plt.show()
```

7.2.2 Modèle en étoile

Ici les données sont organisées autour de plusieurs dimensions. L'agrégation consiste:

- à positionner les données sur des axes (temporels, géographiques,...)
- ou à les organiser de manière hiérarchique en classes (et sous-classes) selon la valeur d'un ou plusieurs attributs.

Exemple de hiérarchie:

- pays > région > département

Dimensions

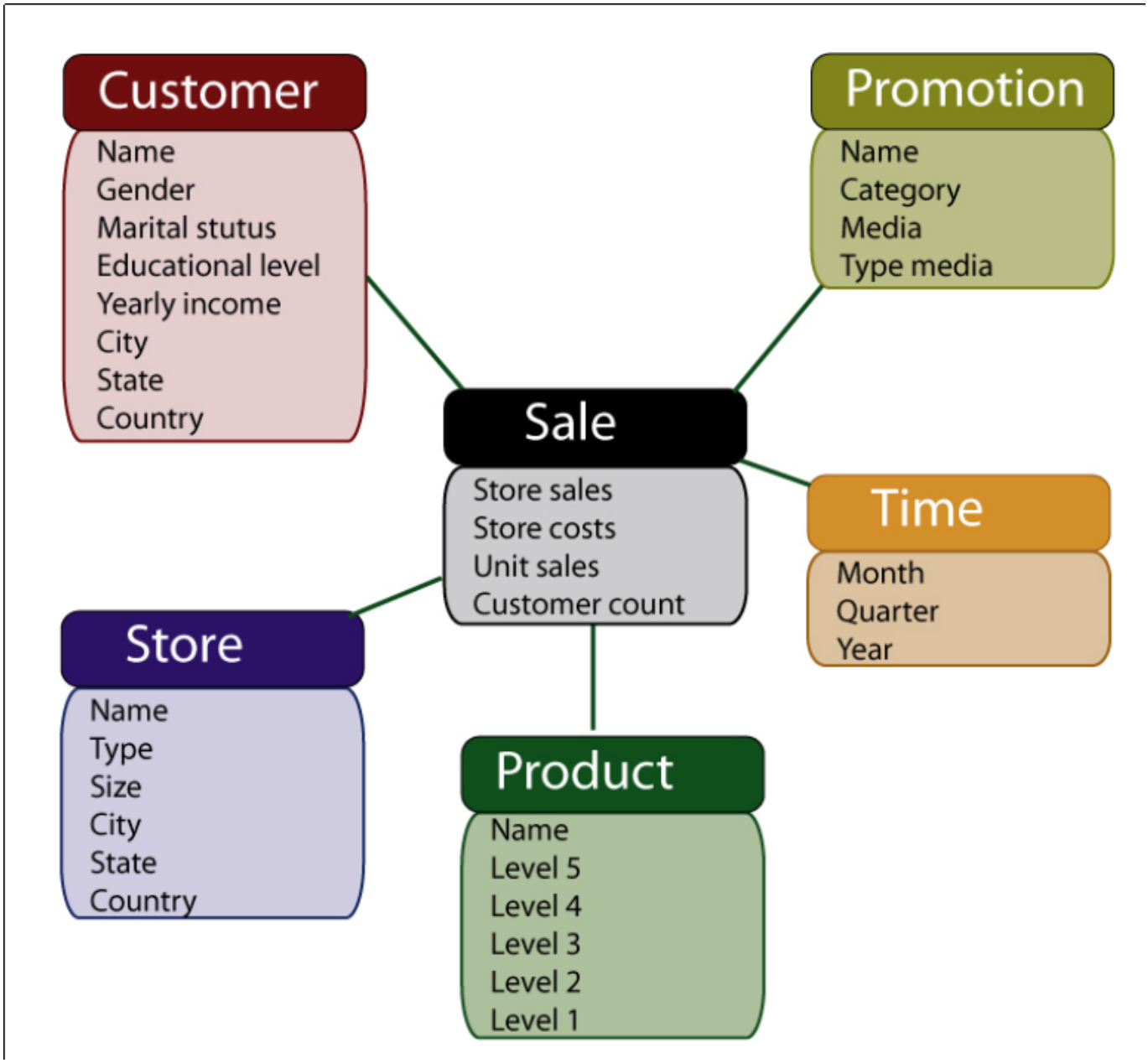
- (qui?) Quels sont les magasins les plus rentables? doit-on ouvrir / fermer des magasins?
- (Où?) répartition des appels/consultation des sites en fonction de l'heure de la journée
- (qui?) Quelle est la liste des clients à contacter?
- (quand?) De quelle quantité doit-on approvisionner quels magasins en fonction de la période de l'année?

Problèmes :

- définir les bons intervalles temporels?? créneaux horaires?
- définir des secteurs géographiques?

Modèle en étoile

- un fait est une association située au centre du schéma. Les attributs de l'association sont les mesures effectuées
- une dimension est une relation participant au fait. Les dimensions sont donc décrites par des attributs (ex : attributs année, trimestre, mois, jour, heure, minute, seconde,...pour une dimension temporelle)
- pour chaque dimension, on décrit une hiérarchie sur les différents attributs de la dimension en définissant un ordre, du particulier au général.



Exemples :

- sur la dimension temporelle : mois ⊂ trimestre ⊂ année
- sur la dimension promotion : nom ⊂ catégorie ⊂ média ⊂ type de média

etc...

Exemples

- Ventes
- Ventes
- Pistes

Cubes de données

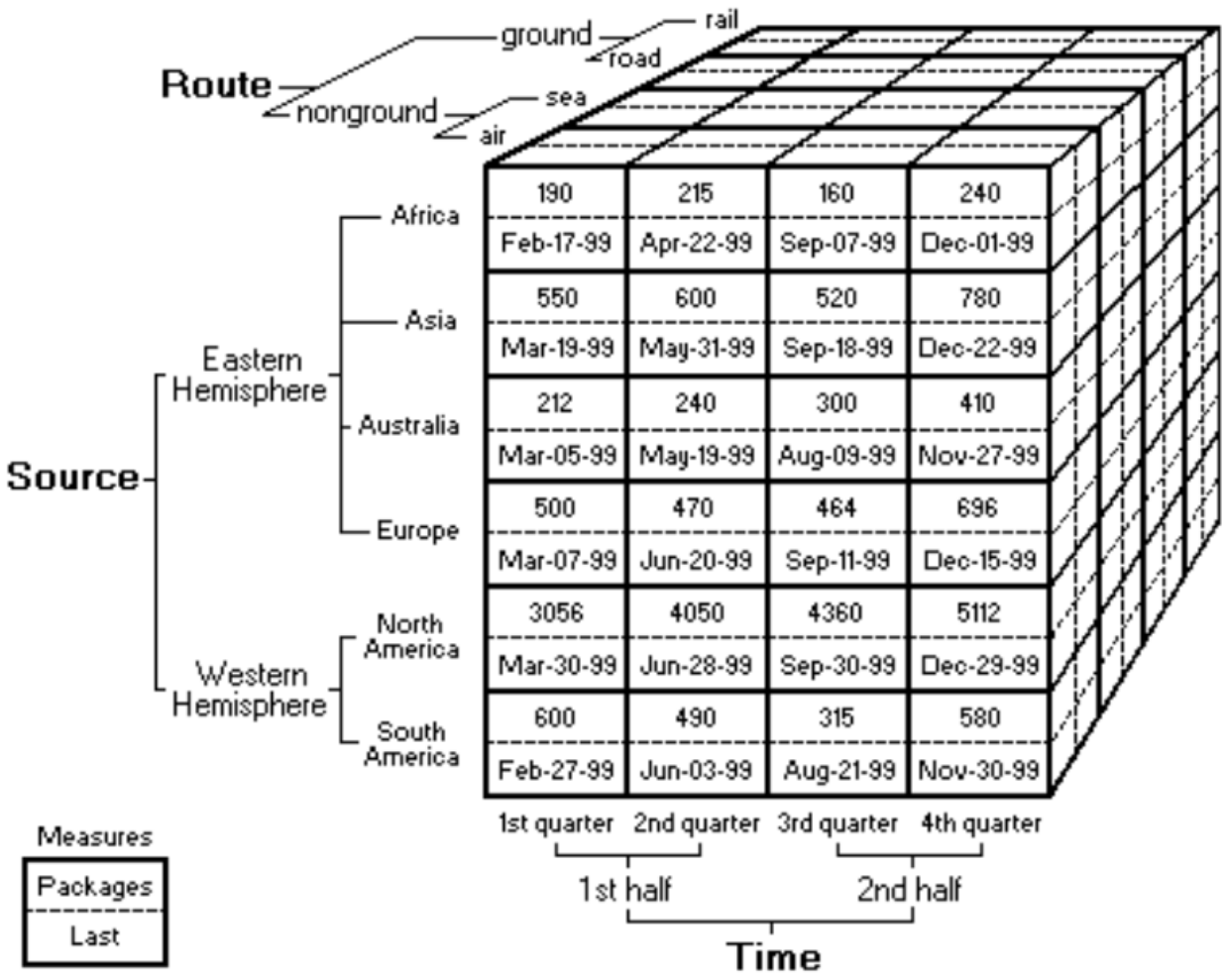
Un cube de données est une structure de données organisée sur le principe des espaces vectoriels. Différents axes sont définis, chaque axe étant associé à une dimension particulière.

- Les dimensions peuvent correspondre à des valeurs discrètes (catégories : type de produit, catégorie de client,…) ou continues (valeurs temporelles ou géographiques, …).
- Chaque fait est décrit comme un point de l'espace vectoriel. Il est positionné dans une cellule du cube. A ce point sont associées une ou plusieur mesures.
- Le cube est un ensemble de cellules (voir figure), chaque cellule correspondant à un intervalle (sur les axes continus) ou une valeur (sur les axes discrets).

Un élément essentiel du modèle de données est la définition de **hiérarchies** sur les dimensions du cube. Chaque dimension se divise en intervalles et sous-intervalles (pour le continu/ quantitatif) ou en catégories et sous-catégories (pour le discret/qualitatif)

Les hiérarchies sur les différentes dimensions permettent de définir le "niveau de résolution" sur les différentes dimensions.

- On peut ainsi s'intéresser à l'évolution d'une certaine grandeur au cours du temps année par année, trimestre par trimestre ou mois par mois selon le niveau de résolution choisi.
- → Hiérarchie : description arborescente d'intervalles et de sous-intervalles sur une dimension. Implemente differentes granularités sur la dimension considérée.



La structure de cube de données est adaptée pour la réalisation d'histogramme multidimensionnels, selon les axes choisis et le niveau de résolution choisi, à l'aide de fonctions d'aggrégation.

- Histogramme et aggrégation
  - (vue quantitative) comptage/répartition d'événements sur un intervalle (discrétisaton d'une distribution d'événements)
  - (vue qualitative) comptage d'événements par catégorie
  - (vue intermediaire) comptage d'événements par catégories hiérarchisées

7.2.2 Mise en oeuvre

XMLA / MDX

7.3 Méthodes avancées

REPRESENTATION DES DONNEES : Représenter des jeux de valeurs de grande taille de façon plus synthétique (algorithmes de réduction de dimension)

REGROUPEMENT ("CLUSTERING") : Définir des regroupements (ou des classements simples ou

hiérarchiques) entre jeux de valeurs

COMPLETION : Méthodes de classification automatique (ou d'interpolation) visant à deviner soit la classe, soit certaines valeurs non mesurées, à partir d'un jeu de valeurs et d'une base d'exemples complets. Il existe des méthodes paramétriques ou non paramétriques.

ESTIMATION ET DECISION : Méthodes visant à estimer la "valeur" associée à un jeu de données (pour l'aide à la décision)