

## A. Introduction

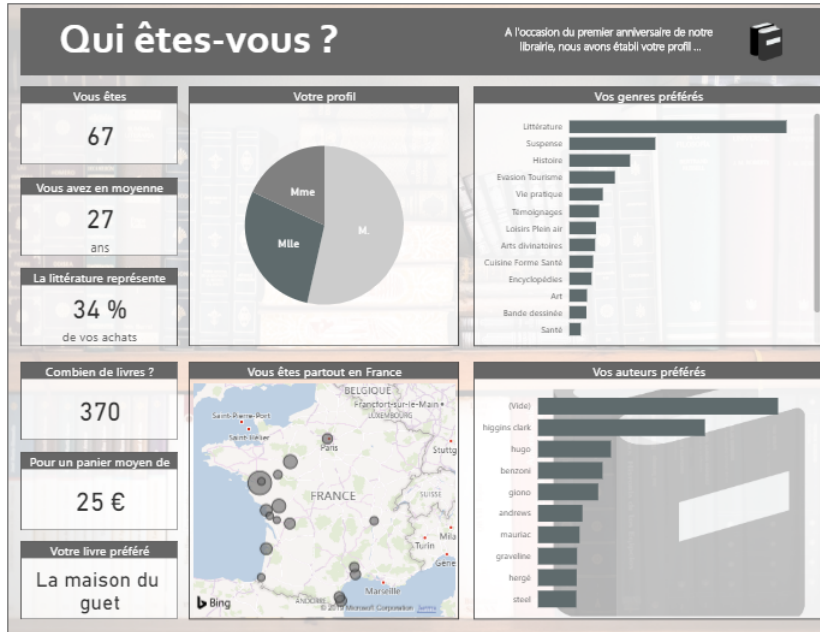
À ce stade, nous avons donc deux fichiers prêts à être exploités.

Le fichier basé sur la source Excel **Ventes Monde.xlsx** va servir dans la première partie de ce chapitre pour découvrir la construction d'un tout premier visuel. Vous pouvez poursuivre sur votre propre fichier ou partir de **Ventes Monde.pbix** disponible en téléchargement.

Dans la deuxième partie, la construction du rapport à travers la découverte d'une variété de visuels est basée sur votre propre fichier **Livres.pbix** ou **Livres - chap3 v1.pbix** disponible en téléchargement.

L'objectif pratique de cette séquence est de réaliser un rapport de profilage des clients.

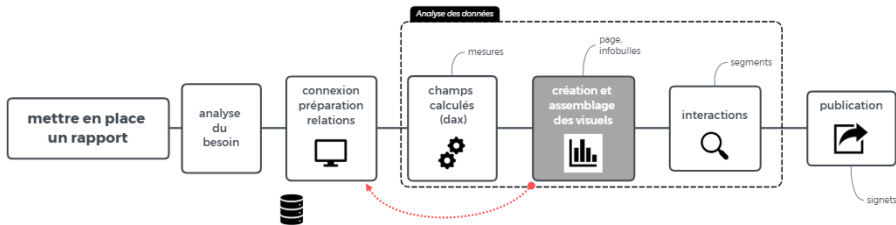
Voici le résultat que nous atteindrons :



Pour cela, nous allons voir les concepts clés, la composition et ses règles, et différents types de visuels.

## B. Les concepts clés

### 1. Le cycle de travail



La mise en place des graphiques et tableaux (les visuels), et de leurs interactions, ne constitue pas l'étape la plus difficile de Power BI, loin de là, et c'est certainement l'étape la plus gratifiante : les données prennent rapidement forme sous une palette de visuels disponibles dans l'outil, complétée par un large choix d'autres visuels disponibles en téléchargement sur la Place de marché.

La construction des visuels peut naturellement amener à créer de nouvelles mesures ou colonnes, et parfois à revenir sur la requête elle-même, de même que les interactions peuvent être mises en place au fur et à mesure : le découpage en trois étapes de l'analyse des données est purement formel.

Dans ce chapitre, nous allons d'une part évoquer les règles générales de construction des visuels, et les règles de construction d'un rapport, et d'autre part découvrir et construire un ensemble de visuels présentés sous forme d'un catalogue de références dans les pages suivantes.

## 2. Fichier, rapport, visuel, interactions et signet

Le *fichier* généré par Power BI porte l'extension .PBIX : il est composé d'un ensemble d'onglets, ou *Pages*, sur lesquelles vous construisez vos *rapports* en assemblant des *visuels*.

Les visuels sont essentiellement de quatre types :

- ▶ Les graphiques (histogramme, carte géographique, par exemple)
- ▶ Les tableaux simples ou croisés (appelés matrice)
- ▶ Les cartes (affichage d'une donnée très résumée, un montant global par exemple)
- ▶ Les segments (des filtres graphiques – cases à cocher, listes – que l'utilisateur du rapport va pouvoir faire varier)

Des objets graphiques (lignes, cadres, images, zones de texte) permettent de compléter et d'habiller le rapport.

Entre les visuels s'opèrent des *interactions* : de manière générale, un visuel peut filtrer ou surligner tout ou partie des autres. Le réglage des interactions peut être modifié pour correspondre exactement au comportement souhaité pour le rapport.

C'est un thème que nous développons dans le chapitre suivant.

Enfin différents états d'un rapport ou différents rapports peuvent constituer une série d'instantanés – les *signets* –, destinés à introduire une forme de narration – raconter les données –, ou à faciliter la navigation. Ce thème est traité dans le dernier chapitre.

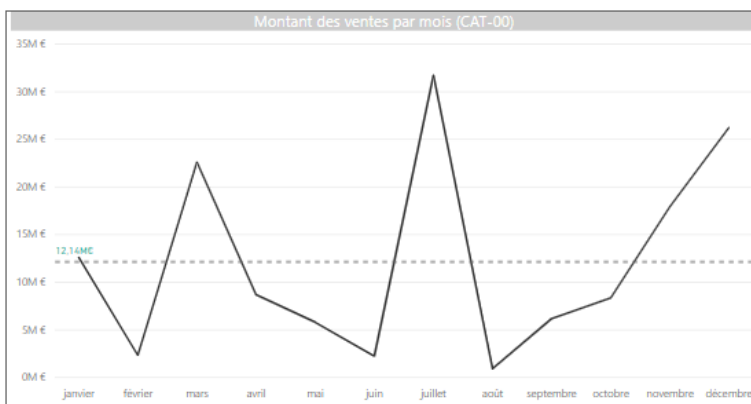
### 3. Mettre en place le premier visuel

La mise en place du visuel a pour objectif de montrer des données pertinentes et justes, elle repose sur les champs choisis et les filtres appliqués, et précède l'étape d'ajustement, où la modification des propriétés du visuel permet de définir exactement son format.

Cette étape est souvent l'occasion de « réduire le bruit », à savoir se débarrasser de toute information qui n'apporte rien de plus : si les données sont directement visibles sur le graphique, les axes d'un graphique ou la légende peuvent s'avérer inutiles, de même les couleurs et les étiquettes.

L'éventail des visuels est très large, et les modalités de leur mise en place sont variables, mais tous répondent aux mêmes règles communes.

Ce graphique, basé sur le fichier **Ventes Monde.pbix**, montre l'évolution du montant mensuel des ventes sur l'année, pour la catégorie **CAT-00** :



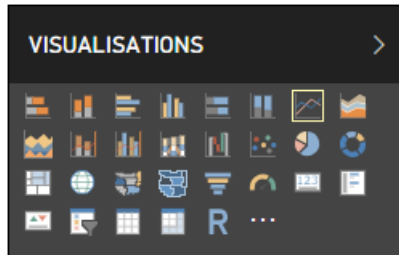
#### Choisir les champs

Trois champs sont requis pour créer ce visuel :

- ▶ La **Date**, dont nous avons vu qu'elle se présente sous forme d'une hiérarchie, de l'année au jour, lorsqu'elle est utilisée.
- ▶ Le **Montant des ventes**.
- ▶ Et la **Catégorie**, invisible sur le graphique, mais utilisée pour filtrer CAT-00.

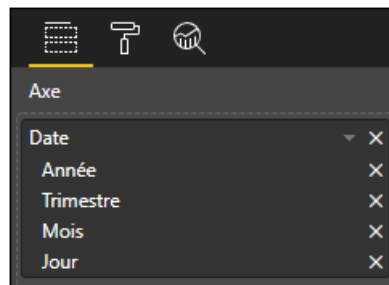
Différentes techniques existent pour créer le visuel à partir du choix des champs, voici la plus rapide :

- ☞ Ouvrez le fichier **Ventes Monde.pbix**.
- ☞ Dans le volet **VISUALISATION**, sélectionnez le type de visuel **Graphique en courbes**.



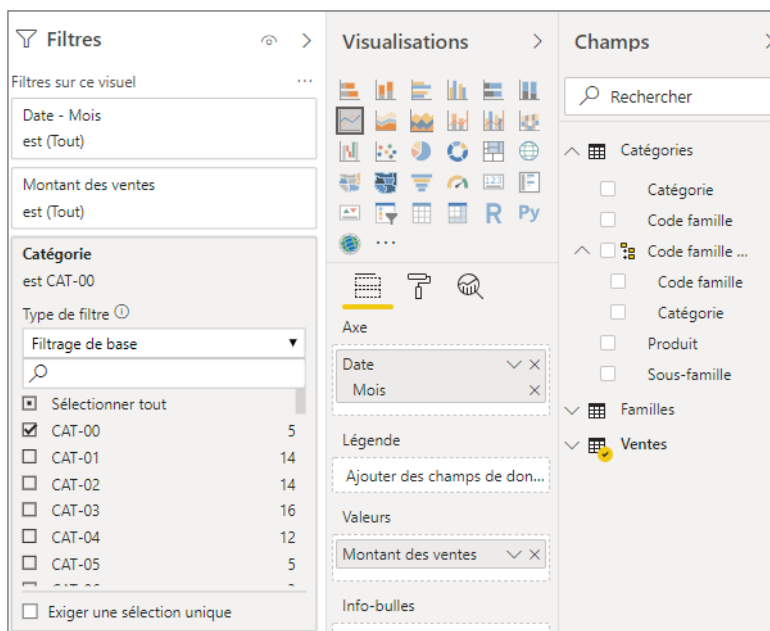
- ☞ Cochez la case **Mois** de la hiérarchie **Date** dans la table **Ventes**.

Si vous ne voyez pas la hiérarchie **Date**, cochez le champ **Date**, et sur le volet **VISUALISATIONS**, dans la zone **Axe**, ouvrez le menu déroulant et supprimez les niveaux inutiles (**Année**, **Trimestre**, **Jour**).

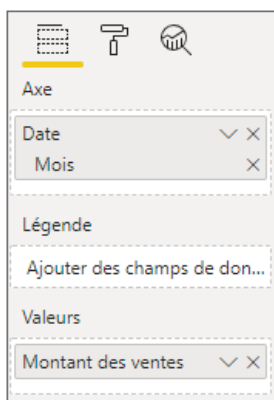


- ☞ Cochez la case **Montant des ventes**.

- Ouvrez le volet **Filtres**, glissez le champ **Catégorie** de la table du même nom dans la zone **Filtres sur ce visuel** et cochez la case **CAT-00** :



Le visuel est en place : regardez maintenant le volet **Champs**.



Chacune des deux données occupe une zone (ou *puits*). Pour beaucoup de visuels, le puits **Axe** (ou son équivalent) est occupé par un ou plusieurs champs textes, codes ou dates, le puits **Valeurs** par un ou plusieurs champs numériques.

## A. Introduction

Le rôle central du langage DAX dans l'analyse des données n'est plus à démontrer : aussi bien pensée que soit votre source de données, aussi bien construit que soit votre modèle, il ne s'agit que des pierres sur lesquelles vous allez pouvoir fonder vos analyses, et celles-ci seront toujours basées sur des formules DAX.

Ce chapitre est centré sur LA fonction DAX par excellence : **CALCULATE**.

Parce que c'est la fonction pivot, celle qui demande de maîtriser les deux notions clés du DAX, à savoir les fonctions de table et le contexte de filtre et, par conséquent de maîtriser toute une batterie d'autres fonctions considérées à juste titre comme centrales.

Et parce que derrière une apparente simplicité – après tout, **CALCULATE** se contente de modifier le contexte de filtre avant de calculer une expression -, se cache un mécanisme d'une richesse et parfois d'une complexité remarquables.

Après un préambule destiné à poser les bases du travail (rappel des notions clés, debugging, validation d'un calcul), nous aborderons **CALCULATE**, nous évoquerons des modèles de formule, nous entrerons dans les arcanes de la fonction, et nous verrons comment **CALCULATE** permet de réaliser des analyses courantes.



Depuis la mise à jour de mai 2020, les séparateurs DAX sont par défaut la virgule pour séparer les listes et les arguments au sein d'une fonction, et le point comme séparateur décimal. Un système « à l'américaine », donc, par opposition aux paramètres localisés, le point-virgule pour séparer les arguments et virgule comme séparateur décimal, pour ce qui concerne la France. Nous restons dans ce livre fidèle à ces derniers. Vous pouvez passer de l'un à l'autre à l'aide du menu **Fichier - Options et paramètres - Options**, puis dans la rubrique **Global - Paramètres régionaux**, faites votre choix des séparateurs DAX :

The screenshot shows the 'Options' dialog box in Power BI Desktop. The left sidebar is expanded to 'Paramètres régionaux' (Regional Settings) under the 'GLOBAL' category. The main content area shows the following settings:

- Langue du modèle**: Utiliser la langue d'affichage par défaut de Windows (dropdown menu).
- Langue utilisée lors de la comparaison de chaînes dans les données et pour créer des champs de date internes**: Utiliser la langue de l'application (dropdown menu).
- Étapes de la requête**: Spécifiez la langue à utiliser pour les noms d'étapes générés automatiquement:
  - Utiliser la langue de l'application
  - Toujours en anglais
- Séparateurs DAX**: Spécifiez la culture à utiliser pour les séparateurs de liste et de décimale dans les expressions DAX:
  - (Recommandé) Utiliser les séparateurs DAX standard : virgule (,) comme séparateur de liste et point (.) comme séparateur décimal
  - Utiliser des séparateurs DAX localisés : les séparateurs de liste et de décimale sont définis par les paramètres régionaux Windows

At the bottom of the 'Séparateurs DAX' section, there is a link: 'En savoir plus'.

*Si vous préférez utiliser les (nouveaux) paramètres recommandés par Power BI, il vous suffira, dans tous les exemples de formules qui suivent, de remplacer le point-virgule par la virgule.*



## B. Préambule



*Cette section traite d'un ensemble de points fondamentaux et généraux de DAX. Vous n'y trouverez donc pas d'exercices d'application, mais je vous encourage à appliquer ces concepts et règles dans toutes vos formules DAX. Il est essentiel que vous lisiez très attentivement les pages qui suivent. Et peut-être, que vous les relisiez par la suite !*

### 1. Les mesures, leur format, leur nom

Pour l'essentiel, les formules doivent être utilisées dans le cadre de la création de mesures, et plus rarement pour la création de colonne.

Rappelons en effet que les mesures ne prennent pas de place dans votre modèle (n'utilise pas d'espace de stockage), et garantissent donc un document plus léger et plus performant. Rappelons que les mesures ne sont calculées qu'au moment de leur utilisation, selon le contexte dans lequel elles apparaissent, et pas pour chaque ligne de la table dans laquelle, à l'inverse, une colonne est « physiquement » créée.

Rappelons également que toute donnée numérique issue de la source et destinée à être analysée (montant des ventes, quantité, résultat de test, etc.), doit être masquée, et remplacée par une mesure équivalente (`[Montant] = SUM(Ventes[Montant des ventes])`). Nous verrons plus loin le lien avec CALCULATE.

Il est par ailleurs utile, lors de la création de formule, de prendre le réflexe de formater tout de suite la mesure nouvellement créée. Enfin pour ce qui concerne le nom de la mesure, évitez les lettres accentuées dans les quelques premiers caractères (afin qu'Intellisense retrouve rapidement votre mesure).

## 2. Le contexte de filtre, le contexte de ligne

Il ne s'agit pas pour nous ici de reprendre intégralement ces notions, mais simplement de rappeler que le contexte de ligne est généré automatiquement lors de la création d'une colonne par le biais d'une formule, ainsi que lors de l'utilisation d'une fonction de type itérateur (en particulier les fonctions `X – SUMX`, etc. – et la fonction `FILTER`).

Le contexte de filtre, lui, est défini par les visuels (tables, graphiques, segments) présents sur le rapport (et parfois sur les autres rapports). Mais il peut aussi être manipulé par formule, et c'est précisément le rôle de la fonction `CALCULATE`.

Le contexte de ligne filtre la table sur laquelle il s'applique, et n'en retient qu'une ligne. L'utilisation des fonctions `RELATED` ou `RELATEDTABLE` permet cependant au contexte de ligne de se propager vers d'autres tables.

Le contexte de filtre *filtre le modèle* dans son ensemble, selon le sens de propagation de 1 à N le long des relations. L'utilisation de la fonction `CROSSFILTER`, ou d'une relation bi-directionnelle, permettent toutefois au filtre de se propager dans le sens N à 1.

Enfin, dans certains cas, le contexte de ligne est transformé en contexte de filtre, selon le principe de la transition de contexte. Nous reviendrons sur cette notion capitale.

## 3. Mise en forme des formules

Afin d'assurer la lisibilité des formules, et notamment lorsqu'elles deviennent complexes, certaines règles de mises en forme peuvent être suivies :

- ▶ Les colonnes référencées le sont toujours en précisant le nom de la table d'abord, alors que les mesures sont simplement référencées entre crochets : cela aide en particulier à mieux « voir » le `CALCULATE` implicite.
- ▶ Ajouter autant de commentaires que nécessaire, précédés d'un `//`.
- ▶ Faciliter la vue des parenthèses ouvrantes et fermantes.
- ▶ Aller à la ligne autant de fois que vous le voulez.

```
[Montant moyen facturé] =  
    AVERAGEX (  
        Date[Date] ;  
        [Montant facturé]  
    )
```

Notez aussi qu'une variante, notamment dans le cadre d'un `CALCULATE`, peut consister à décaler le point-virgule, pour mieux être en mesure de mettre en commentaire une des lignes du code :

```
[Mesure] =
    CALCULATE (
        expression
        ; filtre1
        ; filtre2
        ; filtre3
    )
```

Écrite comme ça, la formule permet de mettre en commentaire facilement (ici, le filtre 3) :

```
[Mesure] =
    CALCULATE (
        expression
        ; filtre1
        ; filtre2
//      ; filtre3
    )
```

#### 4. Raccourcis de l'éditeur de formule DAX

L'éditeur de formule fourmille de raccourcis clavier permettant d'accélérer la saisie ou de faciliter la correction des formules DAX.

Avec un minimum de pratique, vous gagnerez un temps précieux pour remettre en forme, corriger à plusieurs endroits en même temps, intervertir les lignes, etc.

Voici quelques-uns de ces raccourcis.

##### Les cinq raccourcis-clavier les plus utiles

- ▶ Sélectionner toutes les occurrences du terme (ou des caractères) actuellement sélectionné (exemple : une colonne, une table ou une fonction qu'il faut remplacer partout dans la formule).

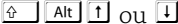


`Ctrl` `F2` (ou `Ctrl` `⇧` `L`)

- ▶ Mettre en commentaire (ou à l'inverse, enlever la mise en commentaire)








Sélectionner la ou les lignes et `Ctrl` `/`

- ▶ Aller à la ligne avec indentation (retrait)








`⇧` `↵`

- ▶ Copier une ligne au-dessus/en-dessous  
 ou 
- ▶ Valider une proposition de fonction d'Intellisense  









### Manipuler les lignes

- ▶ Déplacer une ligne vers le haut/le bas  
 ou 
- ▶ Insérer la ligne ci-dessous  

- ▶ Insérer la ligne ci-dessus  

- ▶ Sélectionner la ligne actuelle  

- ▶ Naviguer jusqu'à une ligne en indiquant son numéro  
 G (taper ensuite  pour masquer la zone de saisie du numéro de ligne)



### Saisir du texte

- ▶ Insérer plusieurs curseurs aux endroits choisis en cliquant (utile pour saisir un même texte à plusieurs endroits)  

- ▶ Activer plusieurs curseurs au même endroit sur plusieurs lignes  
 ou 
- ▶ Sélectionner toutes les occurrences de la sélection actuelle (exemple : une colonne, une fonction qu'il faut remplacer partout).  
 L ou 
- ▶ Sélectionner le mot entier (là où se trouve le curseur) (Si une portion de texte a déjà été sélectionnée,  D sélectionne la prochaine occurrence de la même sélection, et ainsi de suite : cela permet de modifier l'ensemble des portions sélectionnées d'un coup)  


### Commentaire et indentation

- ▶ Mettre en commentaire (ou l'inverse)  
Sélectionner la ou les lignes et  /
- ▶ Augmenter l'indentation / réduire l'indentation  
 / 
- ▶ Aller à la ligne sans indenter  
 
- ▶ Aller à la ligne avec indentation  
 

### Utiliser Intellisense

- ▶ Valider une proposition  

- ▶ Pour rappeler la fenêtre Intellisense  
 I

## 5. Correction de formule avec les variables

Il n'existe pas, dans Power BI, de moteur d'analyse des formules permettant d'en arrêter l'exécution au moment où une erreur est détectée (comme c'est le cas dans Excel notamment). Pour des formules complexes, longues, ceci peut être un problème.

C'est là qu'une utilisation astucieuse des variables peut vous aider.



*Les variables sont déclarées, et exécutées, au début du script. Elles sont couramment utilisées pour n'exécuter un calcul qu'une fois, même si celui-ci est appelé plusieurs fois dans la formule. Leur première utilité est donc d'améliorer la performance de la formule.*

*Leur deuxième raison d'être est d'améliorer la lisibilité de la formule.*

*Leur troisième intérêt est d'effectuer un calcul sur l'état du contexte de filtre au début de l'exécution de la formule (en effet, le calcul n'étant effectué qu'une fois, la valeur de la variable est figée pendant toute la durée de la formule). Ce point, important, est développé à la fin de cette section.*



Pour illustrer ce point, je me sers d'un exemple proposé par Marco Russo dans un tutoriel de Guy In A Cube : <https://www.youtube.com/watch?v=9SV2VnYbgg4>

Nous allons commencer par voir les trois états de mise en forme de la formule : d'abord sans mise en forme, puis avec, et enfin avec l'introduction des variables.

Cette formule retourne une liste de couleurs dont le nombre varie en fonction du choix fait par l'utilisateur à l'aide d'un segment *Couleurs\_seg*.

Premier état :

```
Couleurs =
IF (
    COUNTROWS ( DISTINCT ( 'Produit'[Couleur] ) ) > SELECTEDVALUE
( 'Couleurs_seg'[Nombre de Couleurs] );
    CONCATENATEX ( TOPN ( SELECTEDVALUE ( 'Couleurs_seg'[Nombre de
Couleurs] ); VALUES ( 'Produit'[Couleur] )); 'Produit'[Couleur]; " , " )
    & " et plus...";
    CONCATENATEX ( DISTINCT ( 'Produit'[Couleur] ); 'Produit'[Couleur];
" , " )
)
```

Vous remarquerez tout de suite que le code est très confus, donc difficile à lire, et par conséquent difficile à corriger.

Et en effet, ce code renvoie une erreur. Mais où se situe-t-elle ?

Le premier nettoyage consiste à formater le code selon les règles courantes, par exemple en le passant à la moulinette de [www.daxformatter.com](http://www.daxformatter.com).