

Pilotage Achats – Détection d'anomalies prix (Power BI)

Résumé exécutif

Objectif : analyser des lignes de bons de commande (achats publics SF) pour piloter les **dépenses**, suivre **prix/quantités**, et détecter les **écarts de prix unitaires** (anomalies > 5%).

Livrables : 1 fichier **PBIX** prêt à l'emploi, 1 **PDF** de synthèse (tableau de bord), démo GIF.

Données & outils

- **Dataset** : [*San Francisco Purchasing / Procurement*](#) (CSV local).
- **Périmètre chargé** : échantillon de travail (50–150 dernières lignes) pour prototypage rapide.
- **Outils** : Power BI Desktop (Power Query + DAX).

Tentative API DataSF (Socrata) abandonnée suite à blocages d'auth/ERR_INVALID_RESPONSE.

Contournement : téléchargement CSV local → fiabilisation immédiate.

Processus détaillé

1) Acquisition

1. **CSV local** téléchargé et importé (Accueil → Obtenir des données → Texte/CSV).
2. Pour prototyper : **Conserver les N dernières lignes** (Power Query → Accueil → Réduire les lignes → *Conserver les dernières lignes*).

2) Power Query (ETL) — nettoyage & features

But : colonnes utiles, noms FR, types corrects, calcul du **PU**.

Étapes clés (ordre) :

1. **Choisir les colonnes** (garder l'essentiel) :
Fiscal Year, Purchase Order Date, Purchase Order, Purchase Order Line, Purchasing Department Title, Commodity Title, Supplier & Other Non-Supplier Payees, Encumbered Quantity, Encumbered Amount.
2. **Renommer** (clair & court) :
 - *Fiscal Year* → **AnneeFiscale**
 - *Purchase Order Date* → **DateCommande**
 - *Purchase Order* → **NumeroBC**
 - *Purchase Order Line* → **LigneBC**
 - *Purchasing Department Title* → **DepartementAchat**
 - *Commodity Title* → **LibelleArticle**
 - *Supplier & Other Non-Supplier Payees* → **Fournisseur**
 - *Encumbered Quantity* → **Quantite**

- *Encumbered Amount* → **Montant**

3. **Format :**

- DateCommande : *Date/Heure*
- Quantite, Montant : *Nombre décimal* en remplaçant les « , » par des « . » ou l'inverse en fonction des versions power bi.

4. **PU (Prix unitaire) :**

Ajouter une colonne personnalisée

if [Quantite] <> null and [Quantite] <> 0 then [Montant] / [Quantite] else null
(typer en *Nombre décimal*).

5. **Colonnes temps** (faciliter les visuels) :

- **Année** = Date.Year(DateCommande)
- **Mois** = Date.StartOfMonth(DateCommande)

Problèmes résolus

- “*Nous n’avons pas pu appliquer l’opérateur / aux types Text et Number*” → **cause** : Montant/Quantite en *Texte*. **Fix** : forcer *Nombre décimal* en remplaçant les « , » par des « . » ou l'inverse en fonction des versions power bi AVANT la division.
- *Écran vide / “base disparue”* → **cause** : dernière **étape en erreur**. **Fix** : supprimer l'étape fautive (croix dans *Étapes appliquées*) ou corriger via **Éditeur avancé** (manquait in ou mauvaise dernière référence).

3) Modèle (simple & lisible)

- **Table** : Purchasing_Commodity_Data (fact).
- **Clés analytiques** : NumeroBC, LibelleArticle, Fournisseur, DepartementAchat, AnneeFiscale, Quantite, Montant, Prix unitaire (PU).

4) Mesures DAX

-- Totaux additifs

Dépenses = SUM('Purchasing_Commodity_Data'[Montant])

Quantités = SUM('Purchasing_Commodity_Data'[Quantite])

-- Ratio sûr

PU moyen = DIVIDE([Dépenses], [Quantités])

-- Médiane par Article (affiche BLANK en total pour éviter la confusion)

PU médian article =

IF(

```

    HASONEVALUE('Purchasing_Commodity_Data'[LibelleArticle]),
    MEDIAN('Purchasing_Commodity_Data'[Prix unitaire]),
    BLANK()
)

```

-- Écart de PU en %

ÉcartPU% =

VAR pm = [PU médian article]

VAR pmoy = [PU moyen]

RETURN IF(ISBLANK(pm), BLANK(), DIVIDE(pmoy - pm, pm))

-- Drapeau anomalie (seuil 5 %)

Anomalie =

IF(NOT(ISBLANK([ÉcartPU%])) && ABS([ÉcartPU%]) > 0.05, 1, 0)

-- KPI "BC à risque" (nombre de commandes ayant ≥1 ligne anormale)

BC à risque =

COUNTROWS(

 FILTER(

 VALUES('Purchasing_Commodity_Data'[NumeroBC]),

 CALCULATE(SUMX('Purchasing_Commodity_Data', [Anomalie])) > 0

)

)

-- Montant recalculé (si besoin de fiabiliser les totaux via Q * PU)

Montant recalculé =

SUMX(

 'Purchasing_Commodity_Data',

 'Purchasing_Commodity_Data'[Quantite] * 'Purchasing_Commodity_Data'[Prix unitaire]

)

5) Visuels & mise en page (résultat final)

Page 1 — Synthèse opérationnelle

- **KPI (Cartes)** : Dépenses, Quantités, BC à risque >5%
 - **Barres horizontales : Dépenses par article**
 - Axe : LibelleArticle
 - Valeur : [Dépenses]
 - **Filtre Top N** : *Top 10/15 par Dépenses* (volet Filtres → Top N → “Par valeur” = [Dépenses]).
 - **Scatter : Dispersion Prix vs Volume**
 - X = [Quantités], Y = [PU moyen], Détail = LibelleArticle, Taille = [Dépenses].
 - **Table** (détaillée) avec **MFC** :
 - Colonnes : NumeroBC, LibelleArticle, Fournisseur, Quantite, Montant, Prix unitaire, [PU médian article], [ÉcartPU%].
 - **Filtre visuel** : Anomalie = 1.
 - **Mise en forme conditionnelle** sur [ÉcartPU%] → *Couleur d’arrière-plan* (échelle min=-0.2, centre=0, max=0.2, vert→blanc→rouge).
 - **Slicers** : AnneeFiscale, DepartementAchat, Fournisseur, LibelleArticle (recherche ON).
(Synchronisation possible si plusieurs pages : Afficher → **Synchroniser les segments**).
-

Incidents & correctifs

1. **Crash “base vide” après erreur**
 - *Cause* : dernière **Étape appliquée** invalide → bloque l’aperçu.
 - *Fix* : supprimer l’étape fautive (croix) ou corriger le code via **Éditeur avancé** (ajouter in, pointer l’étape valide).
 2. **Division Texte/Nombre**
 - *Cause* : Montant/Quantite au format *Texte*.
 - *Fix* : **typer Nombre décimal** avant calcul PU (ou *Utiliser la locale* si virgules).
 4. **API DataSF**
 - *Problème* : Auth/pare-feu/ERR_INVALID_RESPONSE.
 - *Fix* : **CSV local** pour avancer et réduire les colonnes.
-

Qualité & validation

- Barre de qualité Power Query = **100 % Valid** sur Quantite, Montant, Prix unitaire.

- Contrôle de cohérence : Montant recalculé \approx Montant d'origine (à \pm arrondis).
 - Vérif anomalie : lignes avec $|\text{ÉcartPU\%}| > 5\%$ ressortent bien **vert/rouge** en MFC.
-

Annexe A — Script M (Power Query)

Colle/remplace dans **Accueil** → **Éditeur avancé** (adapte le chemin CSV).

let

```
Source = Csv.Document(  
    File.Contents("C:\Users\Adem\Downloads\archive\Purchasing_Commodity_Data.csv"),  
    [Delimiter=";", Columns=22, Encoding=1252, QuoteStyle=QuoteStyle.None]  
)
```

```
Promus = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
```

```
// Garder l'essentiel (ajuste si besoin)
```

```
Garder = Table.RemoveOtherColumns(Promus, {  
    "Fiscal Year", "Purchase Order Date", "Purchase Order", "Purchase Order Line",  
    "Purchasing Department Title", "Commodity Title",  
    "Supplier & Other Non-Supplier Payees", "Encumbered Quantity", "Encumbered Amount"  
}),
```

```
// Renommer FR
```

```
Ren = Table.RenameColumns(Garder, {  
    {"Fiscal Year", "AnneeFiscale"},  
    {"Purchase Order Date", "DateCommande"},  
    {"Purchase Order", "NumeroBC"},  
    {"Purchase Order Line", "LigneBC"},  
    {"Purchasing Department Title", "DepartementAchat"},  
    {"Commodity Title", "LibelleArticle"},  
    {"Supplier & Other Non-Supplier Payees", "Fournisseur"},  
    {"Encumbered Quantity", "Quantite"},  
    {"Encumbered Amount", "Montant"}  
}, MissingField.Ignore),
```

```

// Types (si décimales FR : Transformer → Type → Utiliser la locale fr-FR)
Types = Table.TransformColumnTypes(Ren,{
    {"AnneeFiscale", Int64.Type},
    {"DateCommande", type datetime},
    {"NumeroBC", type text},
    {"LigneBC", Int64.Type},
    {"DepartementAchat", type text},
    {"LibelleArticle", type text},
    {"Fournisseur", type text},
    {"Quantite", type number},
    {"Montant", type number}
}),

// Échantillon (prototypage)
Dernieres = Table.LastN(Types, 150),

// PU
PU = Table.AddColumn(Dernieres, "Prix unitaire",
    each if [Quantite] <> null and [Quantite] <> 0 then [Montant]/[Quantite] else null, type number),

// Temps
AjoutAnnee = Table.AddColumn(PU, "Année", each Date.Year(DateTime.Date([DateCommande])),
Int64.Type),
AjoutMois = Table.AddColumn(AjoutAnnee, "Mois", each
Date.StartOfMonth(DateTime.Date([DateCommande])), type date)
in
AjoutMois

```

Annexe B — Mesures DAX

Dépenses = SUM('Purchasing_Commodity_Data'[Montant])

Quantités = SUM('Purchasing_Commodity_Data'[Quantite])

PU moyen = DIVIDE([Dépenses], [Quantités])

PU médian article =

```
IF(
    HASONEVALUE('Purchasing_Commodity_Data'[LibelleArticle]),
    MEDIAN('Purchasing_Commodity_Data'[Prix unitaire]),
    BLANK()
)
```

ÉcartPU% =

VAR pm = [PU médian article]

VAR pmoy = [PU moyen]

RETURN IF(ISBLANK(pm), BLANK(), DIVIDE(pmoy - pm, pm))

Anomalie =

IF(NOT(ISBLANK([ÉcartPU%])) && ABS([ÉcartPU%]) > 0.05, 1, 0)

BC à risque =

```
COUNTROWS(
    FILTER(
        VALUES('Purchasing_Commodity_Data'[NumeroBC]),
        CALCULATE( SUMX('Purchasing_Commodity_Data', [Anomalie]) ) > 0
    )
)
```

Montant recalculé =

```
SUMX(
    'Purchasing_Commodity_Data',
    'Purchasing_Commodity_Data'[Quantite] * 'Purchasing_Commodity_Data'[Prix unitaire]
)
```