

Origine des données INSEE

RGP Communes et IRIS

Les données RGP sont accessibles sur cette [page](#) pour 2025

[Sur cette page pour 2016.](#)

[Sur cette page pour 2011](#)

Il faut télécharger les rubriques Logements et Individus.

-----pour mémoire-----

LE DOSSIER COMPLET NATIONAL

<https://www.insee.fr/fr/statistiques/5359146>

Les dossiers complets par commune (changer code insee)

<https://www.insee.fr/fr/statistiques/2011101?geo=COM-83009>

Pour les fichiers **RGP (1.1 Go chacun)**, filtrer sur PACA (04, 05, 06, 13, 83, 84) avant l'injection est une excellente décision : cela va économiser énormément d'espace disque et accélérer tes requêtes SQL futures.

Comme nous n'avons pas d'outil de filtrage lourd, on va utiliser la puissance native de `awk` sur Debian pour streamer uniquement les lignes dont le code département commence par un de tes départements cibles.

1. Préparation des tables réceptacles

D'abord, on crée les structures vides dans PostGIS.

Bash

```
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "  
CREATE TABLE IF NOT EXISTS insee_raw.rgp_individus_paca_2025 (  
    id serial PRIMARY KEY,  
    donnees jsonb -- On peut stocker en JSONB pour la flexibilité ou définir les colonnes  
);  
CREATE TABLE IF NOT EXISTS insee_raw.rgp_logements_paca_2025 (  
    id serial PRIMARY KEY,  
    donnees jsonb  
);"
```

2. Injection filtrée (Le "Stream")

Voici la commande pour ne garder que la Région PACA. *Note : Je pars du principe que le code département est dans la 2ème ou 3ème colonne. Si tu as un doute sur la position de la colonne, dis-le moi.*

Individus (Filtre Départements 04,05,06,13,83,84) :

Bash

```
awk -F',' 'NR==1 || $2 ~ /^(04|05|06|13|83|84)/' "/home/debian/data/INSEE_RGP_2022-  
25_Individus_zone E.csv" | \  
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo \  
-c "COPY insee_raw.rgp_individus_paca_2025 FROM STDIN WITH (FORMAT CSV, HEADER, DELIMITER ',',  
ENCODING 'UTF8');"
```

Logements (Filtre Départements 04,05,06,13,83,84) :

Bash

```
awk -F',' 'NR==1 || $2 ~ /^(04|05|06|13|83|84)/' "/home/debian/data/INSEE_RGP_2022-  
25_Logements_zone E.csv" | \  
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo \  
-
```

```
-c "COPY insee_raw.rgp_logements_paca_2025 FROM STDIN WITH (FORMAT CSV, HEADER, DELIMITER ',',  
ENCODING 'UTF8');"
```

Pourquoi cette méthode ?

- **Rapidité** : `awk` traite le fichier ligne par ligne sans le charger en mémoire vive.
- **Efficacité** : Seules les lignes utiles traversent le pipe vers Docker.
- **Propreté** : On garde le `HEADER` (la première ligne) pour que la commande `COPY` ne soit pas perdue.

Vérifie juste une chose : Peux-tu me donner les 3 premières lignes d'un des fichiers avec `head -n 3 "/home/debian/data/INSEE_RGP_2022-25_Individus_zone E.csv"` ? Cela me permettra de confirmer le numéro de la colonne département pour ajuster le `$2` dans la commande `awk`.

3- Identification du nom de la colonne

Lance cette commande pour être certain du nom de la colonne (car l'INSEE utilise parfois `COMM` ou `COMMUNE`) :

Bash

```
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "SELECT column_name
```

Bingo ! On a le jackpot. Regarde bien les trois premières lignes du résultat :

1. `commune` : On a bien le code commune (83009 pour Bandol).
2. `arm` : Utile pour Marseille (arrondissements).
3. `iris` : **C'est la surprise du chef.** Tes fichiers RGP contiennent déjà le code IRIS. On n'aura pas besoin d'attendre cet après-midi pour faire de la dentelle géographique.

C'est très clair. La documentation `varmod_indcvi_2022.csv` que tu as partagée se confirme : dans le fichier **Individus**, la colonne `commune` n'existe pas en tant que telle. L'échelle la plus fine disponible ici est `cantville` ou `iris` (si elle apparaît plus bas dans la liste).

Pour le fichier **Logements**, nous avons vu qu'il contient `commune`. Mais pour les **Individus**, nous devons ruser.

1. Vérification de la présence de la colonne `iris`

Avant de filtrer, vérifions si `iris` est présent dans la table Individus (comme c'était le cas pour les Logements) :

Bash

```
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "
SELECT column_name FROM information_schema.columns
WHERE table_name = 'rgp_individus_paca_2025' AND table_schema = 'insee_raw' AND column_name =
'iris';"
```

2. Stratégie d'indexation et d'extraction

Si `iris` existe : On utilise les 5 premiers chiffres de l'IRIS (qui correspondent au code commune).

Bash

```
# Indexation sur IRIS
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "CREATE INDEX
idx_rgp_ind_iris ON insee_raw.rgp_individus_paca_2025 (iris);"

# Extraction Bandol (83009) via IRIS
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "
CREATE TABLE bandol.stats_individus AS
SELECT * FROM insee_raw.rgp_individus_paca_2025
WHERE iris LIKE '83009%';"
```

C'est parfait. La colonne `iris` est présente dans les deux tables. C'est la clé de voûte de notre structuration car elle permet une précision chirurgicale (le quartier) là où `cantville` est trop large.

Voici la séquence de commandes pour finaliser l'extraction de **Bandol** proprement :

1. Indexation sur la colonne `iris` (Individus)

Bash

```
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "CREATE INDEX IF NOT EXISTS idx_rgp_ind_iris ON insee_raw.rgp_individus_paca_2025 (iris);"
```

2. Création de la table Individus pour Bandol

Pour Bandol (code commune 83009), on récupère tous les IRIS commençant par ces 5 chiffres.

Bash

```
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "  
CREATE TABLE bandol.stats_individus AS  
SELECT * FROM insee_raw.rgp_individus_paca_2025  
WHERE iris LIKE '83009%';"
```

3. Création de la table Logements pour Bandol

On utilise la colonne `commune` qui est déjà présente dans cette table.

Bash

```
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "  
CREATE TABLE bandol.stats_logements AS  
SELECT * FROM insee_raw.rgp_logements_paca_2025  
WHERE commune = '83009';"
```

Vérification finale du contenu

Une fois ces commandes passées, on vérifie que les deux tables sont bien peuplées :

Bash

```
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "  
SELECT 'Individus Bandol' as label, count(*) FROM bandol.stats_individus  
UNION ALL  
SELECT 'Logements Bandol' as label, count(*) FROM bandol.stats_logements;"
```

Une fois que tu as ces chiffres, nous sommes prêts pour la première analyse

Nexte_Stats. Par exemple : calculer le **taux de motorisation** des ménages de Bandol (combien ont 0, 1 ou 2 voitures via la variable `voit`) croisé avec leur quartier (`iris`).

Re pour 2016

C'est parfait, les fichiers sont là et font la même taille que ceux de 2022 (~1.1 Go), ce qui confirme qu'on a bien les bases complètes.

Voici le protocole pour intégrer 2016 et préparer l'analyse d'évolution pour **Nexte_Stats**.

1. Création et Import des Individus 2016

On utilise la méthode dynamique pour créer les colonnes automatiquement.

Bash

```
# Création de la table
head -n 1 "/home/debian/data/INSEE_RGP_2016-19_Individus.csv" | sed 's;// text,/g' | sed 's/$/
text/' | xargs -I {} docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c
"CREATE TABLE IF NOT EXISTS insee_raw.rgp_individus_paca_2016 ({});"

# Injection PACA
awk -F';' 'NR==1 || $1 ~ /^(04|05|06|13|83|84)/' "/home/debian/data/INSEE_RGP_2016-
19_Individus.csv" | \
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo \
-c "COPY insee_raw.rgp_individus_paca_2016 FROM STDIN WITH (FORMAT CSV, HEADER, DELIMITER ';',
ENCODING 'UTF8');"

```

2. Création et Import des Logements 2016

Bash

```
# Création de la table
head -n 1 "/home/debian/data/INSEE_RGP_2016-19_Logements.csv" | sed 's;// text,/g' | sed 's/$/
text/' | xargs -I {} docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c
"CREATE TABLE IF NOT EXISTS insee_raw.rgp_logements_paca_2016 ({});"

# Injection PACA
awk -F';' 'NR==1 || $1 ~ /^(04|05|06|13|83|84)/' "/home/debian/data/INSEE_RGP_2016-
19_Logements.csv" | \
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo \
-c "COPY insee_raw.rgp_logements_paca_2016 FROM STDIN WITH (FORMAT CSV, HEADER, DELIMITER ';',
ENCODING 'UTF8');"

```

3. Extraction Bandol 2016

Maintenant, on isole Bandol (83009). On vérifie si les colonnes sont les mêmes (`iris` pour les individus, `commune` pour les logements).

Bash

```
# Extraction Individus 2016
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "
CREATE TABLE bandol.stats_individus_2016 AS
SELECT * FROM insee_raw.rgp_individus_paca_2016
WHERE iris LIKE '83009%';"

# Extraction Logements 2016
docker exec -i alteris_postgis psql -U alteris_admin -d alteris_geo -c "
CREATE TABLE bandol.stats_logements_2016 AS
SELECT * FROM insee_raw.rgp_logements_paca_2016
WHERE commune = '83009';"
```

Pourquoi c'est génial pour ton projet :

Une fois ces commandes terminées, on pourra lancer la requête de la "Vérité" : **L'évolution de la population réelle de Bandol en 6 ans.**

Lance ces imports (compte environ 5 minutes pour l'ensemble). Dès que c'est fini, on compare les chiffres de 2016 avec les 8 268 habitants de 2022 !

Revision #1

Created 2026-04-10 09:15:16 UTC by Julien

Updated 2026-04-10 09:15:16 UTC by Julien