

# Tables nécessaires

---

Les tables qu'il vous faut pour lancer tous les scripts dans ce dossier sont les suivants:

- **SUPPORT.csv**
- **EMETTEUR.csv**
- **getPopCodePostal.csv**

Pour obtenir les tables **SUPPORT.csv** et **EMETTEUR.csv** il faut suivre la documentation que l'autre groupe avait fait.

Pour obtenir **getPopCodePostal.csv** il faut lancer le script **getPop.py** dans le dossier **\SociauxAntenne\script**.

## Ordre d'exécution des scripts pour obtenir les graphiques

---

Tous les scripts se trouvent dans **\SociauxAntenne\script\_stat\_generaion\_population**. Aucun n'as de paramètre lors de l'exécution.

Le seul script qu'il faut **lancer en premier** est **creationDossierStatistiques.py**.

Il permet de créer l'arborescence de dossiers dans lesquels graphiques en .png seront sauvegardés.

Mis à part cela, vous pouvez lancer les scripts suivants dans l'ordre que vous voulez:

- **statGen.py**
- **statGenParOperateur.py**
- **evolutionGenGlobalTous.py**
- **evolutionGenGlobalAncien.py**
- **evolutionGenRegionTous.py**
- **evolutionGenRegionAncien.py**
- **evolutionGenRegionRecent.py**

## Spécificités des scripts

---

### Afficher les graphiques lors du lancement des scripts

---

Pour afficher les graphiques lors du lancement des scripts il vous suffit d'ajouter cette ligne à la fin des scripts:

```
plot.show()
```

NB: certains scripts créent plus d'un graphique à la fois donc vous pouvez avoir plusieurs fenêtres qui s'ouvrent.

# statGen.py

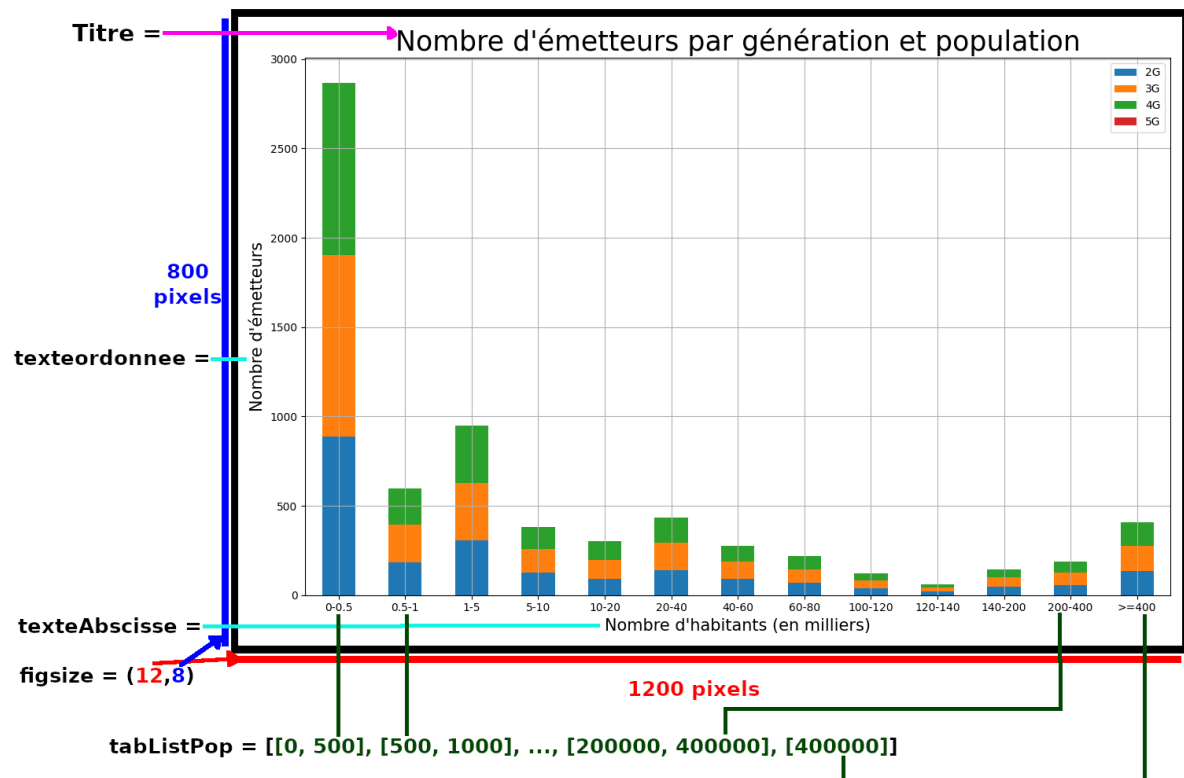
crée deux graphiques dans le dossier  
/statistiques/emetteur\_population\_support/StatToutOperateurConfondu.  
Le premier graphique, statPopGenTous.png compte une fois chaque génération  
présente sur chaque support sans prendre en compte l'opérateur.  
Le second, statPopGenMax.png représente la génération maximale de tous les  
émetteurs présents sur chaque support sans prendre en compte l'opérateur.

## Variables modifiables

Les variables suivantes peuvent être modifiées sans que cela n'affecte le bon fonctionnement du programme:

```
tabListPop = [[0, 500], [500, 1000], [1000, 5000], [5000, 10000], [10000, 20000], [20000, 40000], [40000, 60000], [60000, 80000], [100000, 120000], [120000, 140000], [140000, 200000], [200000, 400000], [400000]]
figsize = (12, 8)
tailleTitre = 25
titre = "Nombre d'émetteurs par génération, population et opérateur"
titreMax = "Nombre d'émetteurs de génération maximale\npar population et opérateur"
tailleOrdonnee = 15
tailleAbscisse = 15
texteAbscisse = "Nombre d'émetteurs"
texteOrdonnee = "Nombre d'habitants (en milliers)"
```

## Exemple de graphique généré par statGen.py avec explication des variables



# À quoi correspondent ces variables ?

## tabListPop

tableau d'intervalles de population. Les intervalles de population sont sur l'axe des abscisses. **La dernière valeur est obligatoire et n'est pas une paire.** Elle correspond au dernier intervalle qui correspond à toutes les régions dont le nombre d'habitants est supérieur à 400 000 habitants de base.

**Vous pouvez avoir des intervalles qui ne se suivent pas** comme :

```
tabListPop = [[1, 2], [9, 10], [11]]
```

**Les valeurs entre ces deux intervalles ne sont pas pris en compte.** Si vous voulez les valeurs entre il suffit d'ajouter l'intervalle qui correspond aux valeurs entre les deux.

## figsize

Correspond aux **dimensions du graphique en centaines de pixels.**

Si vous voulez un graphique qui sur une image de taille 1200\*1000 il vous suffit de changer **figsize** par ceci:

```
figsize = (12,10)
```

## tailleTitre, titre et titreMax

Ces trois variables sont claires mais au cas où, voici ce qu'elles représentent:

- **tailleTitre:** Correspond à la taille de police des deux titres des deux graphiques créés par statGen.py
- **titre:** Correspond au titre du premier graphique créé par statGen.py (statPopGenTous.png)
- **titreMax:** Correspond au titre du second graphique créé par statGen.py (statPopGenMax.png)

## tailleOrdonnee, tailleAbscisse, texteOrdonnee, texteAbscisse

Également, ces 4 variables représentent bien leur noms:

- **tailleOrdonnee** et **tailleAbscisse:** Représente la taille de la police des textes descriptifs de l'axe des abscisses et des ordonnées
- **texteOrdonnee** et **texteAbscisse:** Représente le texte descriptif des abscisses et des ordonnées

# statGenParOpérateur.py

---

Il crée dans le dossier

/statistiques/emetteur\_population\_support/StatParOperateur 8 graphiques, 2 pour chaque opérateur:

- statPopGenMax[Opérateur].png: représente la génération maximale par émetteur de chaque support de l'opérateur [Opérateur]
- statPopGenTous[Opérateur].png: représente tous les émetteurs de chaque support de chaque support de l'opérateur [Opérateur]

Il crée ensuite dans le dossier

/statistiques/emetteur\_population\_support/StatParOperateurMemeOrdonne 10 graphiques, 2 par opérateur et 2 comprenant tous les opérateurs.

Les 8 graphiques sont nommés comme ceux créés précédemment et utilisent les mêmes données mais la hauteur de l'ordonnée est la même pour tous.

La hauteur est le nombre maximal (par tranche de population) d'émetteurs de l'opérateur qui possède le plus d'émetteurs toute génération confondus.

Les deux autres graphiques sont:

- statPopGenOperateurTous.png: représente tous les opérateurs et toutes les générations par support sur un seul graphique.
- statPopGenOperateurGenMax.png: représente tous les opérateurs et la générations maximale de leur support sur un seul graphique.

Ces deux derniers graphiques sont peu lisibles c'est pourquoi les 8 premiers sont créés pour comparer les opérateurs en nombre entre eux.

## Variables modifiables

```
tabListPop = [[0, 500], [500, 1000], [1000, 5000], [5000, 10000], [10000, 20000], [20000, 40000], [40000, 60000], [60000, 80000], [100000, 120000], [120000, 140000], [140000, 200000], [200000, 400000], [400000]]
figsize = (12, 8)
tailleTitre = 25
titreGraphCommun = "Nombre d'émetteurs par génération, population et opérateur"
titreGraphCommunMax = "Nombre d'émetteurs de génération maximale\npar population et opérateur"
titreFreeMax = "Nombre d'émetteurs de génération maximale\nde Free Mobile"
titreFree = "Nombre d'émetteurs par génération et population \nde Free Mobile"
titreSFRMax = "Nombre d'émetteurs de génération maximale\nde SFR"
titreSFR = "Nombre d'émetteurs par génération et population \nde SFR"
titreOrangeMax = "Nombre d'émetteurs de génération maximale\nd'Orange"
titreOrange = "Nombre d'émetteurs par génération et population \nd'Orange"
titreBouyguesMax = "Nombre d'émetteurs de génération maximale\nde Bouygues Télécom"
titreBouygues = "Nombre d'émetteurs par génération et population \nde Bouygues Télécom"
tailleOrdonnee = 15
tailleAbscisse = 15
texteAbscisse = "Nombre d'émetteurs"
texteOrdonnee = "Nombre d'habitants (en milliers)"
```

## À quoi correspondent ces variables ?

### Variables similaires à statGen.py

Toutes les variables correspondent à ce qu'elle représente dans le script **statGen.py**.

Les seules différences minimales sont les suivantes:

- **taille[variable]:** Change la taille de ce que la variable représente pour les 18 graphiques
- **titreGraphCommun** et **titreGraphCommunMax:** Correspond aux titres des graphiques où tous les opérateurs sont présents.
- **titre[Opérateur]** et **titre[Opérateur]Max:** Titres des 16 autres graphiques. Chaque titre est utilisé deux fois. La première fois où ils sont utilisés c'est pour la représentation du nombre d'antennes proportionnel à l'opérateur. La deuxième fois est pour les graphiques dont l'ordonnée est celle de l'intervalle de population de l'opérateur qui possède le plus d'émetteurs.

# evolutionGenGlobalTous.py

Crée deux graphiques dans le dossier  
/statistiques/emetteur\_population\_support/StatParAnnee.

Sur l'axe des ordonnées on a les intervalles de populations.  
Sur l'axe des abscisses on a le nombre d'émetteurs totale mis en service ou le pourcentage correspondant à l'intervalle de population.

Les barres de l'histogramme correspondent aux différents intervalles d'années.  
La hauteur des barres correspond au nombre d'émetteurs mis en service durant cet intervalle d'années et pour l'intervalle de population en question.

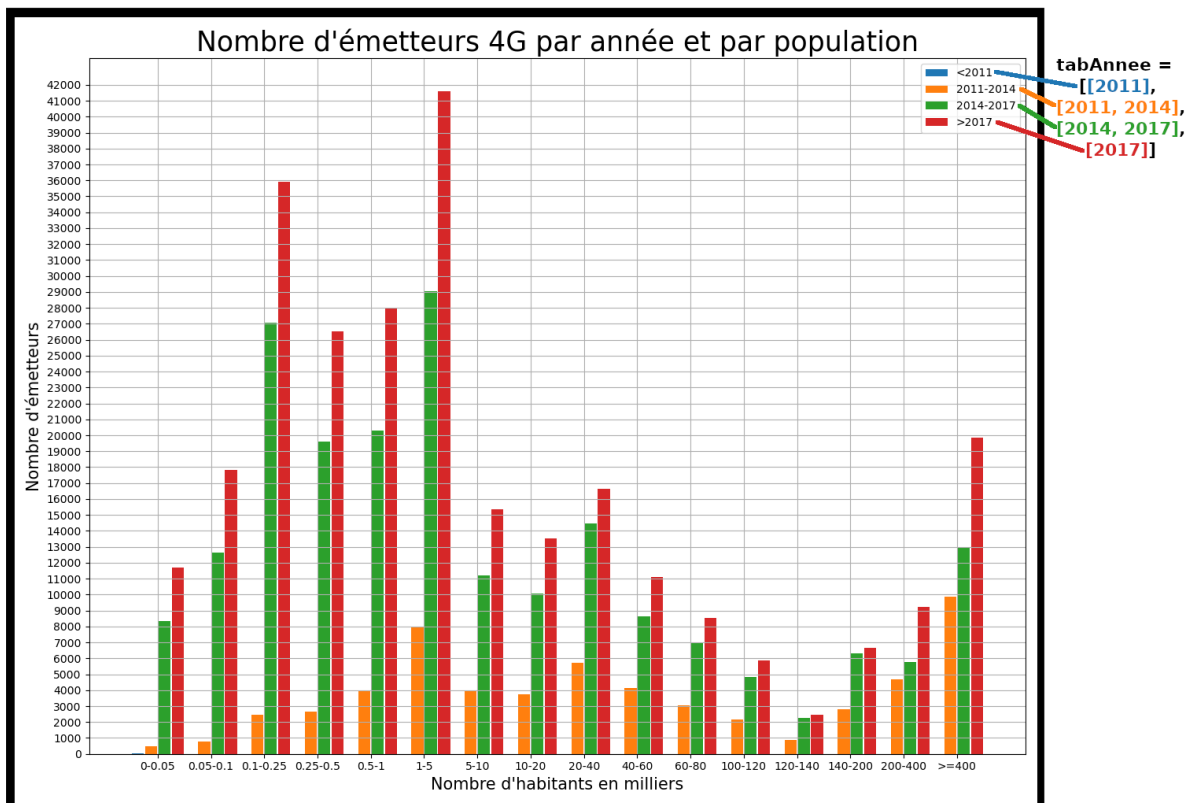
Les deux graphiques sont les suivants:

- statAnneeGlobal.png: Représente en nombre pour chaque intervalle de population le nombre d'émetteurs mis en service par rapport au total de l'intervalle de population en question.
- statAnneeGlobalPourcentage.png: Représente en pourcentage pour chaque intervalle de population le pourcentage d'émetteurs mis en service par rapport au total de l'intervalle de population en question.

## Variables modifiables

```
tabListPop = [[0, 50], [50, 100], [100, 250], [250, 500], [500, 1000], [1000, 5000], [5000, 10000], [10000, 20000], [20000, 40000], [40000, 60000], [60000, 80000], [100000, 120000], [120000, 140000], [140000, 200000], [200000, 400000], [400000]]
tabAnnee = [[2011], [2011, 2014], [2014, 2017], [2017]]
generation = "LTE"
figsizeNombres = (13, 10)
figsizePourcent = (15, 10)
titre = "Nombre d'émetteurs 4G par année et par population"
titrePourcentage = "Pourcentage d'émetteurs 4G par année et par population"
tailleTitre = 25
texteOrdonneeGlobal = "Nombre d'émetteurs"
texteAbscisseGlobal = "Nombre d'habitants en milliers"
texteOrdonneePourcentage = "Pourcentage d'émetteurs"
texteAbscissePourcentage = "Nombre d'habitants en milliers"
tailleOrdonnee = 15
tailleAbscisse = 15
```

## Exemple de graphique généré par evolutionGenGlobalTous.py avec explication des variables



# À quoi correspondent ces variables ?

## Variables différentes

### **generation**

Correspond à la génération d'émetteurs que l'on veut étudier. Dans la table EMETTEUR.csv "LTE" correspond à la 4G.

Cette variable permettra d'obtenir les mêmes graphiques pour la 5G quand les opérateurs commenceront à installer leur émetteurs ou bien pour les générations précédentes si l'on prend les bons intervalles d'années.

### **tabAnnee**

Correspond à un tableau de tableau qui correspond aux intervalles d'années que l'on souhaite étudier.

**La première** et **la dernière** valeur ne sont pas des paires.

La première correspond à l'intervalle pour lesquels la date de mise en service de l'émetteur est inférieur strictement à la première valeur.

La dernière correspond de manière analogue à l'intervalle qui contient tous les émetteurs mis en service après la dernière valeur

## Variables similaires à statGen.py

**tabListPop**, **figsizeNombres**, **figsizePourcent**, **titre**, **titrePourcentage**, **tailleTitre**, **texteOrdonneeGlobal**, **texteAbscisseGlobal**, **texteOrdonneePourcentage**, **texteAbscissePourcentage**, **tailleOrdonnee** et **tailleAbscisse** sont exactement les mêmes que pour **statGen.py** dont la seule différence est qu'ils sont liés à différents graphiques pour les titres.

Ici toutes les variables qui n'ont pas Pourcentage en fin sont liés au graphique qui représente en nombre les émetteurs mis en service durant les différents intervalles de temps.

Toutes les variables qui ont Pourcentage en fin sont liés au graphique qui représente en pourcentage les émetteurs mis en service durant les différents intervalles de temps et proportionnellement à chaque intervalle de population.



# evolutionGenGlobalAncien.py

Crée deux graphiques de manière similaire à evolutionGenTous.py dans le même dossier (/statistiques/emetteur\_population\_support/StatParAnnee).

La seule différence est qu'ici on ne compte qu'un émetteur par support et on ne prend que celui avec la date de mise en service la plus vieille.

Les deux graphiques sont les suivants:

- statAnneeGlobalAncien.png
- statAnneeGlobalPourcentageAncien.png

## Variables Modifiables

```
tabListPop = [[0, 50], [50, 100], [100, 250], [250, 500], [500, 1000], [1000, 5000], [5000, 10000], [10000, 20000], [20000, 40000], [40000, 60000], [60000, 80000], [100000, 120000], [120000, 140000], [140000, 200000], [200000, 400000], [400000]]
tabAnnee = [[2011], [2011, 2014], [2014, 2017], [2017]]
generation = "LTE"
figsizeNombres = (13, 10)
figsizePourcent = (15, 10)
titre = "Nombre d'émetteurs 4G par année et par population"
titrePourcentage = "Pourcentage d'émetteurs 4G par année et par population"
tailleTitre = 25
texteOrdonneeGlobal = "Nombre d'émetteurs"
texteAbscisseGlobal = "Nombre d'habitants en milliers"
texteOrdonneePourcentage = "Pourcentage d'émetteurs"
texteAbscissePourcentage = "Nombre d'habitants en milliers"
tailleOrdonnee = 15
tailleAbscisse = 15
```

## À quoi correspondent ces variables ?

C'est exactement les mêmes que pour evolutionGenGlobalTous.py.

Tout ce qui change est le traitement de données.

# evolutionGenRegionTous.py

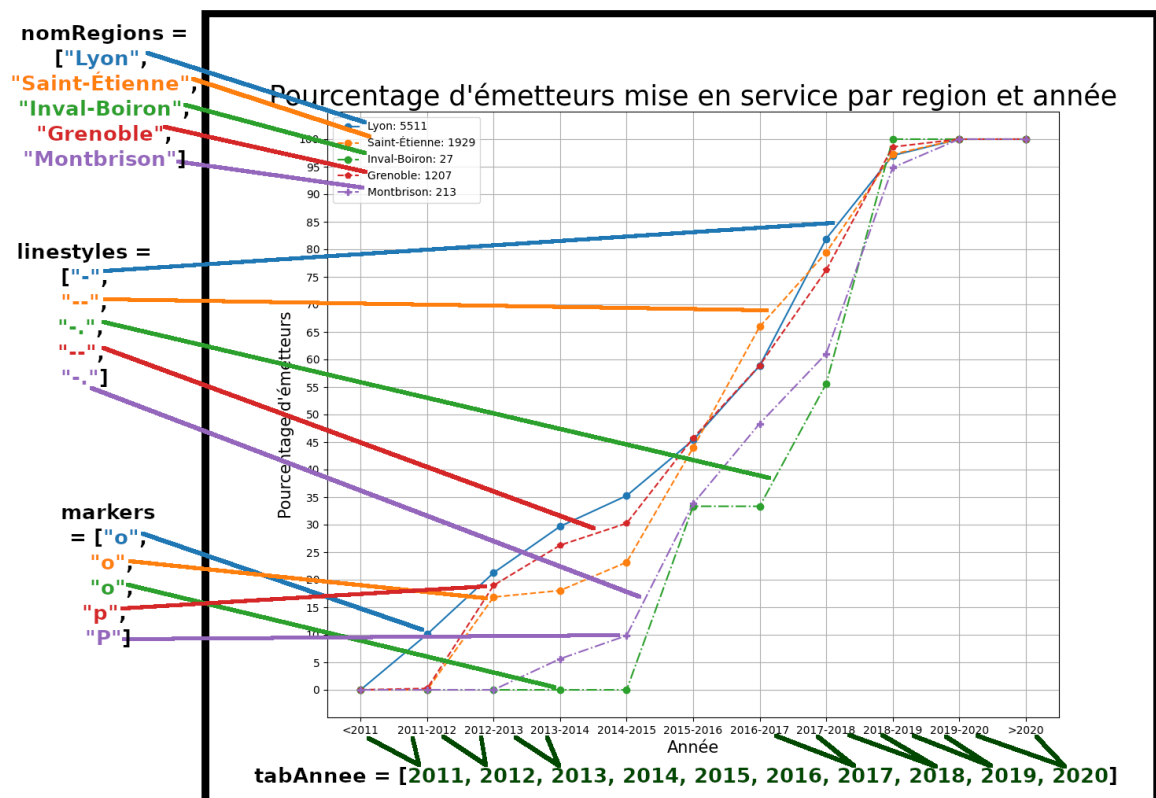
Crée un graphique dans le dossier  
/statistiques/emetteur\_population\_support/StatParAnnee.

Le graphique est cumulatif et représente pour chaque région l'évolution par année de leur couverture réseau en pourcentage.  
ici on compte chaque émetteur présent sur un support par région.

## Variables modifiables

```
generation = "LTE"
figsize = (12, 10)
titre = "Pourcentage d'émetteurs mise en service par region et année"
tailleTitre = 25
axeOrdonnee = "Pourcentage d'émetteurs"
axeAbscisse = "Année"
tailleOrdonnee = 15
tailleAbscisse = 15
tabAnnee = [2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020]
regions = [[69001, 69002, 69003, 69004, 69005, 69006, 69007, 69008, 69009],
[42000, 42100, 42230], [80430], [38000,38100,38700], [42600]]
nomRegions = ["Lyon", "Saint-Étienne", "Inval-Boiron", "Grenoble", "Montbrison"]
linestyles = ["-", "--", "-.", "---", "-."]
markers = ["o", "o", "o", "p", "p"]
texteSauvegarde = "statRegions.png"
```

## Exemple de graphique généré par evolutionGenRegionTous avec explication des variables



# À quoi correspondent ces variables ?

## Variabiles similaires aux autres graphiques

Toutes les variables au noms similaires aux autres scripts font exactement la même chose

## Variable différentes

- **tabAnnee:** représente les intervalles d'année sur l'axe des abscisses. La **première** et la **dernière** valeurs du tableau sont utilisé pour deux intervalles.
- **texteSauvegarde:** correspond au nom du graphique qui sera sauvegardé dans le dossier /statistiques/emetteur\_population\_support/StatParAnnee
- **nomRegions:** représente le nom des régions, arrondissements ou même juste villes que l'on souhaite étudier.
- **regions:** tableau de tableau qui contient les code postaux des régions que l'on souhaite étudier.
- **linestyles:** représente le style de ligne qui sera dessiné par le graphique pour chaque région. [Plus d'information sur les linestyles disponibles.](#)
- **markers:** représente le style de marker qui sera placé sur chaque ligne à chaque année. [Plus d'information sur les markers disponibles.](#)

**nomRegions, regions, linestyles** et **markers** sont liés. La i-ème case d'un tableau est liés à l'i-ème case des deux autres tableaux.

Si vous voulez par exemple représenter l'évolution de la couverture du 1er, 2ème et 3ème arrondissements de Lyon il vous suffit de faire:

```
regions = [[69001], [69002], [69003]]
nomRegions = ["Lyon 1er", "Lyon 2ème", "Lyon 3ème"]
linestyles = ["-", "--", "-."]
markers = ["o", "o", "o"]
texteSauvegarde = "statRegionsLyon.png"
```

**NB:** Pensez à changer le nom du fichier de sauvegarde en changeant **texteSauvegarde** pour ne pas perdre les graphiques que vous allez générer si vous souhaitez étudier plusieurs régions différentes.

# evolutionGenRegionAncien.py

Crée un graphique dans le dossier  
/statistiques/emetteur\_population\_support/StatParAnnee.

Le graphique est similaire à celui créé par `evolutionGenRegionTous.py`. Il cumule et représente pour chaque région l'évolution par année de leur couverture réseau en pourcentage.  
La différence est qu'ici on ne prend en compte que la date la plus ancienne par émetteur présent sur le support et par région.

## Variables modifiables

```
generation = "LTE"
figsize = (12, 10)
titre = "Pourcentage d'émetteurs mise en service par region et année"
tailleTitre = 25
axeOrdonnee = "Pourcentage d'émetteurs"
axeAbscisse = "Année"
tailleOrdonnee = 15
tailleAbscisse = 15
tabAnnee = [2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020]
regions = [[69001, 69002, 69003, 69004, 69005, 69006, 69007, 69008, 69009],
[42000, 42100, 42230], [80430], [38000,38100,38700], [42600]]
nomRegions = ["Lyon", "Saint-Étienne", "Inval-Boiron", "Grenoble", "Montbrison"]
linestyles = ["-", "--", "-.", "---", "-."]
markers = ["o", "o", "o", "p", "P"]
texteSauvegarde = "statRegionsEmetAncien.png"
```

## À quoi correspondent ces variables ?

C'est exactement les mêmes que pour `evolutionGenRegionTous.py`. Seul le traitement des données change.