|  |
| --- |
| Enrichissement géomarketing |
| Documentation module |
| *Le 01/06/2017* |



**SOMMAIRE**

[1 Introduction 3](#_Toc484085921)

[2 Geocodage 3](#_Toc484085922)

[2.1 Utilisation 3](#_Toc484085923)

[2.2 Sortie 4](#_Toc484085924)

[3 Enrichissement âge 5](#_Toc484085925)

[3.1 Utilisation 5](#_Toc484085926)

[3.2 Sortie 6](#_Toc484085927)

[4 Enrichissement géomarketing 6](#_Toc484085928)

[4.1 Utilisation 6](#_Toc484085929)

[4.2 Sortie 7](#_Toc484085930)

[5 Profils 8](#_Toc484085931)

[5.1 Utilisation 8](#_Toc484085932)

[5.2 Sortie 10](#_Toc484085933)

# Introduction

Le présent document a pour but d’expliquer les différentes étapes pour développer un module d’enrichissement geomarketing .

Le module est composé d’un package python, d’un programme exemple de lancement et de maquettes Excel utilisées pour les profils de populations.

Le module est disponible nommé INEnrichissement, tourne en python 3.12.

L’enrichissement comporte 4 étapes, 3 d’entre elles étant totalement indépendantes :

* Le géocodage
* L’enrichissement de l’age
* L’enrichissement des informations géomarketing
* Les profils de populations, qui ne peuvent être lancés qu’après l’exécution d’un enrichissement des informations geomarketing

La suite du document va détailler les différentes étapes.

# Geocodage

## Utilisation

Le géocodage va permettre de récupérer 2 informations, à partir de l’adresse :

* **Le code INSEE** : identifiant unique des communes françaises
* **Le code IRIS** : niveau de découpage géographique le plus fin auquel l’INSEE fournit des informations

La fonction python du package INEnrichissement à utiliser est la fonction « EG\_Insee\_Iris ».

Les arguments de la fonction sont les suivants :

* table\_entree : ***Obligatoire***, table de données, au format table
* top\_TNP : ***Obligatoire***, 1 si les champs civilite, nom et prénom sont dans le même champ dans la table de données, 0 si les champs sont dans des champs distincts, au format integer
* civilite : nom du champ contenant la civilité
* prenom : nom du champ contenant le prénom
* nom : nom du champ contenant le nom
* complement\_nom : nom du champ contenant le complément du nom
* adresse : ***Obligatoire***, nom du champ contenant l’adresse
* complement\_adrs : nom du champ contenant le complément d’adresse
* lieu\_dit : nom du champ contenant le lieudit
* cp : ***Obligatoire***, nom du champ contenant le code postal
* ville : ***Obligatoire***, nom du champ contenant la ville
* id\_client : ***Obligatoire***, nom du champ contenant l’identifiant client
* pays : nom du champ contenant le pays
* email : nom du champ contenant l’adresse email
* tel : nom du champ contenant le numéro de téléphone

Attention, si un paramètre facultatif est vide (par exemple, il n’y a pas de champ contenant le prénom), l’argument ne doit pas être mis dans l’appel de la fonction :

testRNVP =

Attention, lors de la première utilisation du module, une fenêtre MSDOS noire peut demander une confirmation de transfert de fichiers. Il suffit de taper « yes » dans la fenêtre puis de valider. Cette confirmation ne sera demandée que lors du premier lancement.

## Sortie

L’objet de sortie est une table de données au format table, contenant les champs initiaux, ainsi que 4 champs supplémentaires :

* **c\_insee** : code insee de l’adresse sur 5 caractères
* **c\_iris** : code iris de l’adresse sur 4 caractères
* **c\_qualite\_iris** : qualité de l’enrichissement IRIS (1 : correspondance exacte, 2 : correspondance partielle, 8 : non trouvé ou non disponible)
* **codgeo** : code géographique utilisé par l’enrichissement, sur 9 caractères (concaténation du code Insee et du code Iris)

# Enrichissement âge

## Utilisation

L’enrichissement de l’âge va permettre d’estimer l’âge d’un client en fonction de son prénom et de son lieu d’habitation.

La fonction python du package INEnrichissement à utiliser est la fonction « EG\_age\_sexe ».

Les arguments de la fonction sont les suivants :

* **tb\_client** : table de données, au format table
* **prenom** : nom du champ contenant le prénom
* **sexe** : nom du champ contenant le sexe, obligatoirement au format « F » / « M », mettre NA si champ absent
* **age\_declare** : nom du champ contenant l’âge déclaré, mettre NA si champ absent
* **top\_estim\_sexe** : 1 si le sexe doit être estimé à partir du prénom, 0 sinon
* **codgeo** : nom du champ contenant le code géographique (rappel : 9 caractères, concaténation des codes INSEE et IRIS si IRIS présent, de INSEE et 0000 si IRIS absent)
* **ajust** : 1 si un ajustement de l’age doit être fait, 0 sinon
* **var\_ajust** : nom de la variable par rapport à laquelle faire l’ajustement, NA si pas de variable d’ajustement

Le principe de l’ajustement est le suivant : après avoir estimé l’âge de l’ensemble des clients de la base, on va faire une seconde passe qui va ajuster chaque âge estimé en fonction des autres âges estimés sur la base. Imaginons qu’on estime l’âge d’un client à 65 ans, mais que la moyenne d’âge de la base est de 25 ans, alors on va réajuster l’âge estimé pour qu’il se rapproche de la moyenne. Cet ajustement est facultatif, et peut également être fait selon un critère (par exemple le sexe, qui fait qu’on ajustera l’âge estimé d’un client en fonction de l’âge estimé sur les clients du même sexe)

Contrairement à la fonction de géocodage, tous les arguments de la fonction doivent être présents dans l’appel de la fonction, avec NA pour les arguments non remplis :

tb\_client = testRNVP,

prenom = "prenom",

sexe = NA,

age\_declare = NA,

top\_estim\_sexe = 1,

codgeo ="codgeo",

ajust = 0,

var\_ajust = NA)

## Sortie

L’objet de sortie est une table de données au format table, contenant les champs initiaux, ainsi que 6 champs supplémentaires :

* e\_sexe : sexe estimé (si demandé)
* e\_annee\_naissance : année de naissance estimée
* e\_p\_5ans : probabilité que l’âge réel soit dans une fourchette de +/- 5 ans de l’âge estimé
* indice\_conf\_age : indice de confiance de l’âge estimé (de Confiance - - à Confiance + +)
* e\_top\_age\_ok : indicateur d’estimation de l’âge (1 : âge déclaré, 2 : âge estimé, 3 : pas d’estimation)
* e\_age : âge estimé

# Enrichissement géomarketing

## Utilisation

L’enrichissement geomarketing va permettre d’estimer certaines données socio-démographiques (PCS, situation familiale, niveau d’études, statut d’habitation, nombre d’enfants, niveau de revenus) ainsi que d’ajouter certaines données liées au lieu d’habitation (nombre d’habitants, typologie de commune, typologie commerce et typologie logement)

La fonction python du package INEnrichissement à utiliser est la fonction « EG\_Enrichissement\_Geomk ».

Les arguments de la fonction sont les suivants :

* table\_initiale : table de données, au format table
* codgeo : nom du champ contenant le code géographique(rappel : 9 caractères, concaténation des codes INSEE et IRIS si IRIS présent, de INSEE et 0000 si IRIS absent)
* var\_sexe : nom du champ contenant le sexe, obligatoirement au format « F » / « M »
* var\_age : nom du champ contenant l’âge

Il n’y a pas d’argument facultatif à cette fonction.

Exemple d’appel :

testGeomk =

L’objet de sortie est une table de données au format table, contenant les champs initiaux, ainsi que les champs supplémentaires suivant :

* e\_PCS : Catégorie socio professionnelle estimée
* c\_indice\_qualite\_pcs : indice de qualité de l’estimation de la PCS
* e\_situation\_fam : situation familiale estimée
* c\_indice\_qualite\_menage : indice de qualité de l’estimation de la situation familiale
* e\_etudes : niveau d’études estimé
* c\_indice\_qualite\_formation : indice de qualité de l’estimation du niveau d’études
* h\_ind : probabilité d’habiter en habitation individuelle
* locat\_hlm : probabilité d’habiter en HLM
* e\_habitat\_individuel : classe d’habitation individuelle estimée
* e\_habitat\_hlm : classe d’habitation HLM estimée
* e\_statut\_hab : statut d’habitation estimé
* c\_indice\_qualite\_logement : indice de qualité de l’estimation du statut d’habitation
* e\_proba\_1\_enfant : classe de probabilité d’avoir au moins 1 enfant
* e\_proba\_2\_enfants : classe de probabilité d’avoir au moins 2 enfants
* e\_proba\_m5 : classe de probabilité d’avoir 1 enfant de moins de 5 ans
* e\_proba\_5\_10 : classe de probabilité d’avoir 1 enfant âgé entre 5 et 10 ans
* e\_proba\_10\_15 : classe de probabilité d’avoir 1 enfant âgé entre 10 et 15 ans
* e\_proba\_15\_20 : classe de probabilité d’avoir 1 enfant âgé entre 15 et 20 ans
* e\_typo\_commune\_2010 : typologie de la commune de résidence
* e\_taille\_commune : taille de la commune de résidence
* e\_seg\_commerces : segmentation commerce du lieu d’habitation
* e\_sous\_seg\_commerces : sous segmentation commerce du lieu d’habitation
* e\_seg\_logement : segmentation logement du lieu d’habitation
* i\_rev : revenu annuel estimé
* e\_revenus : classe de revenu estimé
* e\_decile : décile de revenu estimé
* c\_indice\_qualite\_rev : indice de qualité de l’estimation du revenu
* e\_dept : département de résidence
* e\_region\_9 : « grande » région de résidence
* e\_Region : région de résidence
* e\_lib\_dept : libellé du département de résidence
* e\_reg : numéro de région de résidence

# Profils

## Utilisation

Le profil va permettre de visualiser les résultats de l’enrichissement. 3 types de profils sont disponibles :

* Comparaison d’une population cible vs population totale
* Comparaison de 2 populations
* Comparaison de la base avec la population globale d’une zone géographique

La fonction python du package INEnrichissement à utiliser est la fonction « EG\_profil ».

Les arguments de la fonction sont les suivants :

* **sortie** : ***Obligatoire***, chemin dans lequel doivent être extraites les sorties
* **table** : ***Obligatoire***, table de données issue de l’enrichissement, au format table
* **var\_age** : ***Obligatoire***, nom du champ contenant l’âge
* **var\_sexe** : ***Obligatoire***, nom du champ contenant le sexe, au format « F » / « M »
* **var\_pop1** : nom de la variable sur laquelle faire la restriction de population cible, à ne remplir que pour la comparaison d’une population cible avec la population totale ou pour la comparaison de 2 populations
* **modalite\_pop1** : valeur de la variable sur laquelle faire la restriction de population cible, à ne remplir que pour la comparaison d’une population cible avec la population totale ou pour la comparaison de 2 populations
* **var\_pop2** : nom de la variable sur laquelle faire la restriction de population cible 2, à ne remplir que pour la comparaison de 2 populations
* **modalite\_pop2** : valeur de la variable sur laquelle faire la restriction de population cible 2, à ne remplir que pour la comparaison de 2 populations
* **code\_geo** : nom de la variable sur laquelle faire la restriction de population géographique, à ne remplir que pour la comparaison géographique – ne peut être que « e\_dept », « e\_iris » ou « e\_insee »
* **zone\_geo** : valeur de la variable sur laquelle faire la restriction de population géographique, à ne remplir que pour la comparaison géographique

Les arguments qui ne servent pas ne doivent pas être déclarés lors de l’appel des fonctions

Exemples d’appels :

* Comparaison d’une population cible vs population totale

test\_profil\_1 =

* Comparaison de 2 populations

test\_profil\_2 =

* Comparaison de la base avec la population globale d’une zone géographique

test\_profil\_3 =

## Sortie

Les sorties sont un fichier csv extrait dans le répertoire spécifié, puis dans un sous répertoire spécifique à chaque type de profil. Ce csv alimente ensuite une maquette Excel à placer dans le même sous dossier.

* Comparaison d’une population cible vs population totale

Le sous-dossier s’appelle « Profil cible » et la maquette « 2016\_modele\_profil\_cible.xls »

Un exemple est disponible dans le répertoire « Sorties\Exemples » du module : 20170601\_profil\_cible\_exemple.xls

* Comparaison de 2 populations

Le sous-dossier s’appelle « Profil cible1 VS cible2 » et la maquette « 2016\_modele\_profil\_cible\_vs\_cible.xls »

Un exemple est disponible dans le répertoire « Sorties\Exemples » du module : 20170601\_profil\_cible\_vs\_cible\_exemple.xls

* Comparaison de la base avec la population globale d’une zone géographique

Le sous-dossier s’appelle « Profil zone geo » et la maquette « 2016\_modele\_profil\_zone\_geo.xls »

Un exemple est disponible dans le répertoire « Sorties\Exemples » du module : 20170601\_profil\_zone\_geo\_exemple.xls