|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Index | Nom | Type before | Type after | Nombre valeurs | % NA | Valeurs | Code NA |
| 1 | **Id** | int | int | 300 000 | 0 | 1 : 300.000 |  |
| 2 | **annee\_naissance** | int |  | 68 | 0.097 | 1925 - 1998 |  |
| 3 | **annee\_permis** | int |  | 69 | 0.097 | 1944 - 2018 |  |
| 4 | **Marque** | char | char | 155 | 0.097 | Peugeot, … |  |
| 5 | **puis\_fiscale** | int | num | 51 | 0 | 0 - 58 |  |
| 6 | **anc\_veh** | int | num | 44 | 0 | 0 - 99 |  |
| 7 | **codepostal** | char | char | 23711 | 0.0017 | 69719 |  |
| 8 | **energie\_veh** | char | char | 5 | 0.138 | ‘gazole’ |  |
| 9 | **kmage\_annuel** | num | num | 1244 | 0 | 282 – 17.000 |  |
| 10 | **crm** | int |  | 149 | 0 | 50 – 270 |  |
| 11 | **profession** | char | char | 17 | 0 | ‘ouvrier’ |  |
| 12 | **var1 (nombre annees permis)** | int |  | 68 | 0.097 | -2 – 72 | nan |
| 13 | **var2** | int |  | 6 | 0 | 0 : 5 |  |
| 14 | **var3** | int | char | 2 | 0 | 272/372 |  |
| 15 | **var4** | int |  | 10 | 0 | 0 : 9 |  |
| 16 | **var5** | int | char | 2 | 0 | 1 : 2 |  |
| 17 | **var6** | char |  | 5 | 0 | A/B/C/D/N |  |
| 18 | **var7** | int |  | 18 | 0.097 | 0 – 50 ‘NR’ | NR |
| 19 | **var8** | char | char | 24 | 0.195 | ‘B1’, ‘H1’, … |  |
| 20 | **var9** | int |  | 335 | 0 | 0 – 620 |  |
| 21 | **var10** | int |  | 213 | 0 | 0 – 548 |  |
| 22 | **var11** | int |  | 44 | 0 | 0 : 42 / 99 |  |
| 23 | **var12** | int |  | 1244 | 0 | 8 – 3211 |  |
| 24 | **var13** | int |  | 4 | 0 | 2 : 5 |  |
| 25 | **var14** | char | char | 8 | 0 | ‘N’, ‘E’, ‘D’ |  |
| 26 | **var15** | int |  | 10 | 0 | 0 : 9 |  |
| 27 | **var16** | int |  | 6 | 0.110 | 1, 3, ‘NR’, … | NR |
| 28 | **var17** | int |  | 6 | 0 | 0 : 5 |  |
| 29 | **var18** | int |  | 46 | 0 | 0 : 45 |  |
| 30 | **var19** | int |  | 29 | 0 | 0 - 32 |  |
| 31 | **var20** | int | char | 2 | 0 | 0/1 |  |
| 32 | **var21** | int | char | 2 | 0 | 0/1 |  |
| 33 | **var22** | int |  | 5 | 0 | 0 - 60 |  |

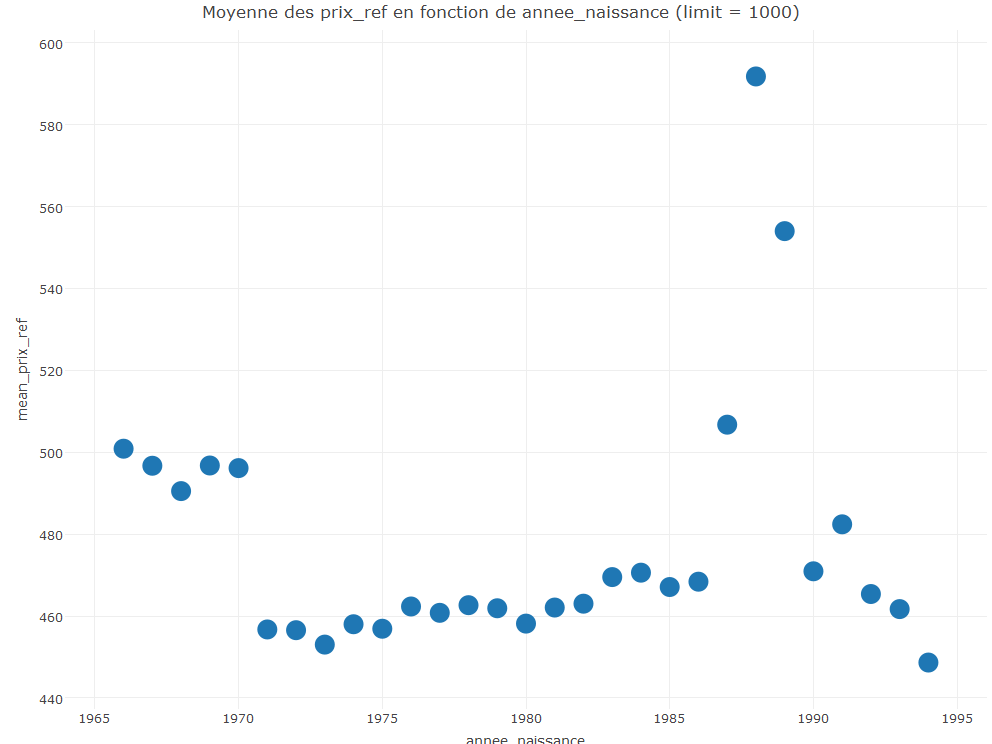
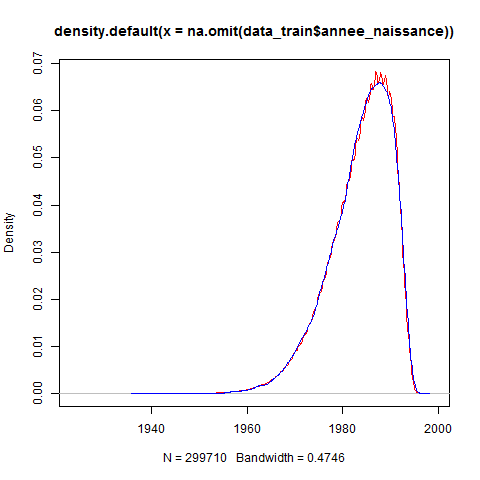
* On a vu que var12 et kilometrage sont tres coréllées : une seule des deux si random forrest (comportement identique)
* Déterminer quelles variables mettre en char et lesquelles en int
* Regarder les corrélations entre variable (pairs + corr)
* Code postal : calculer la valeur moyenne par code postal/ département et déterminer un facteur +/\* au modèle construit sans le code postal
* Analyser variable par variable si on les passe en quanti/quali
* Regarder si Train et Test ont la même distribution ( valeurs de test non présentes dans test, …)

# Croisement variables Devis Maif :

* Date début assurance
* 2 : Déjà assuré Maif : OUI / NON
* 5 : Type véhicule : Auto/ Moto / Cyclomoteur / Camping-car / Quad
* 155 : Marque
* 18 : puissance fiscale
* X : modele
* X : ancienneté véhicule
* 5 : énergie véhicule
* N : nom véhicule
* 4 : nombre de portes : 2 / 3 / 4 / 5
* 2 : usage : privé / professionnel
* 4 : conducteur principal : vous / conjoint / concubine / enfant a charge
* 3 : passe assuré : assuré ailleurs / assuré occas maif / sans expérience
  + Ailleurs :
    - 2 : Sinistres 24 dernier mois
    - 2 : Suspension annulation permis 2 derniers mois
    - N : coefficient de bonus malus
    - N : date de dernière échéance contrat actuel
  + Sans expérience :
    - 2 : conduite accompagnée avec la maif ?
* 2 : sexe : mme / mr
* X : date permis (année)
* X : date naissance (année)

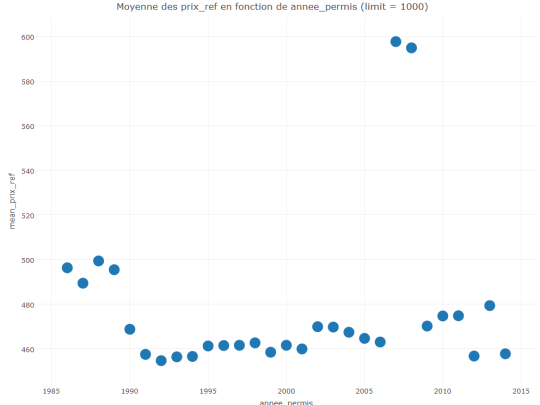
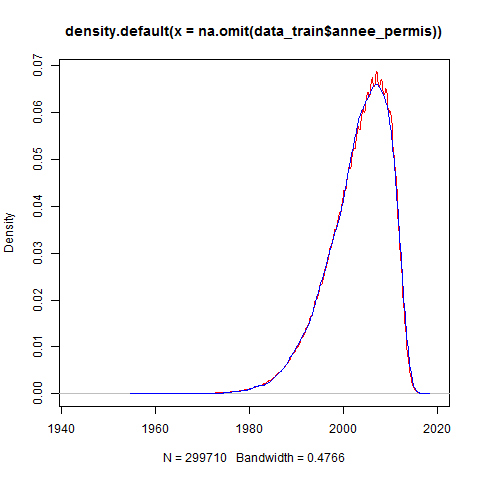
# Analyse des variables

## Année naissance :



* Repartition test/Train : OK
* Valeur bizarre en 1988 et 1989
* A enlever ?

## Année permis :

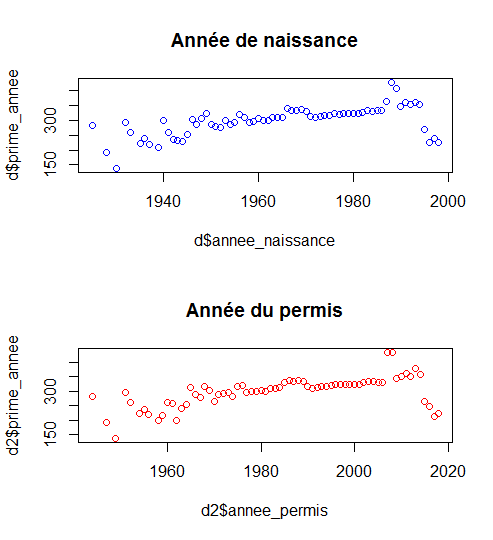


* IDEM que précedement

## Annee naissance et année de permis:

Si on regarde l’année de naissance et l’année de permis, ces variables sont très corrélées. Une régression linéaire de la date de permis en fonctions de la data de naissance donne : permis = 1.00000 \* naissance + 18.83.

Ces variables d’années ne semblent pas avoir d’influence simple à définir sur le prix de l’assurance. On peut donc créer d’autres variables à partir de ces dernières :



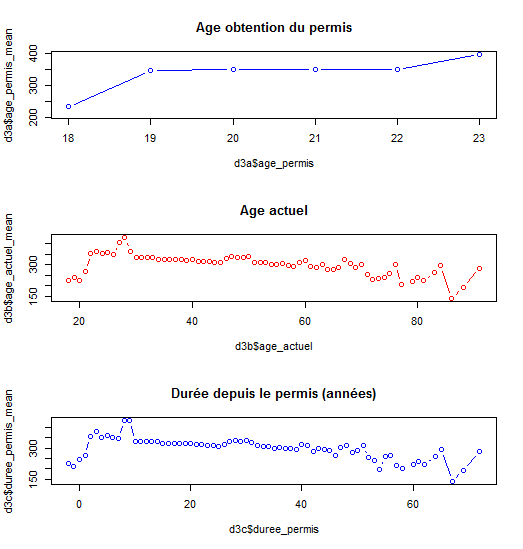
Il nous faudrait un moyen de recréer ces données : par exemple :

* Annee de naissance + age du permis
* Age du permis + age actuel + age depuis permis

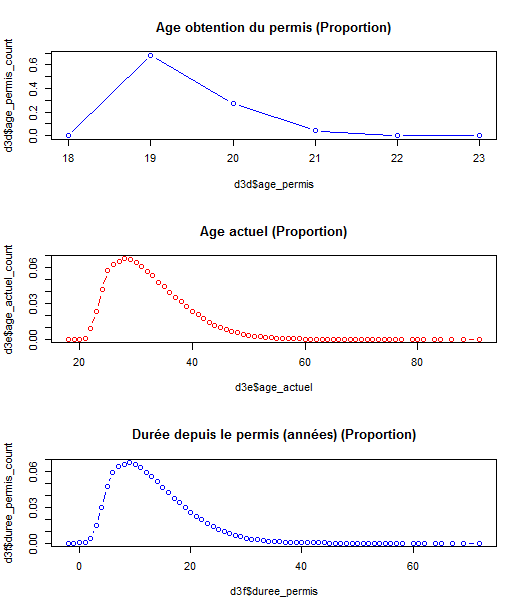
On fait le test de créer les variables :

* Age lors de l’obtention du permis (année\_permis – année\_naissance)
* Age actuel
* Durée depuis le permis (2016 – année permis) => VAR 1 !!

On obtient les valeurs moyennes suivantes pour l’assurance :



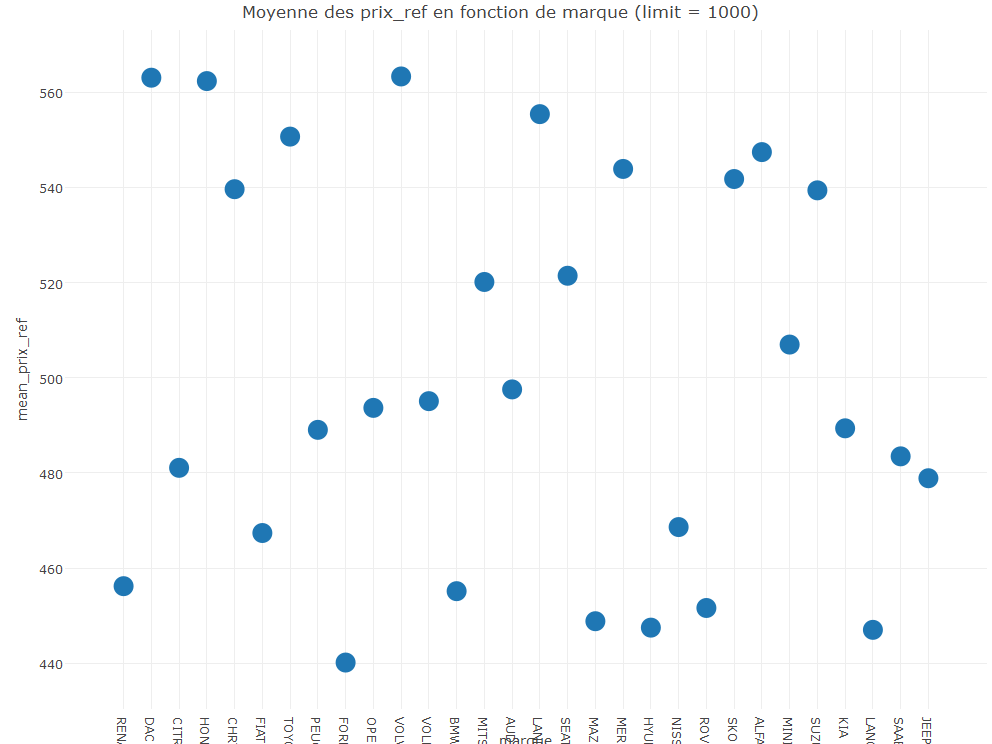
Et les effectifs suivant en proportions :



Il semblerait pertinent de remplacer « annee\_naissance » et « annee\_permis » par « age\_permis » et une des deux variables créées parmi (« age\_actuel » et « duree\_depuis\_permis », qui semblent très corrélées).

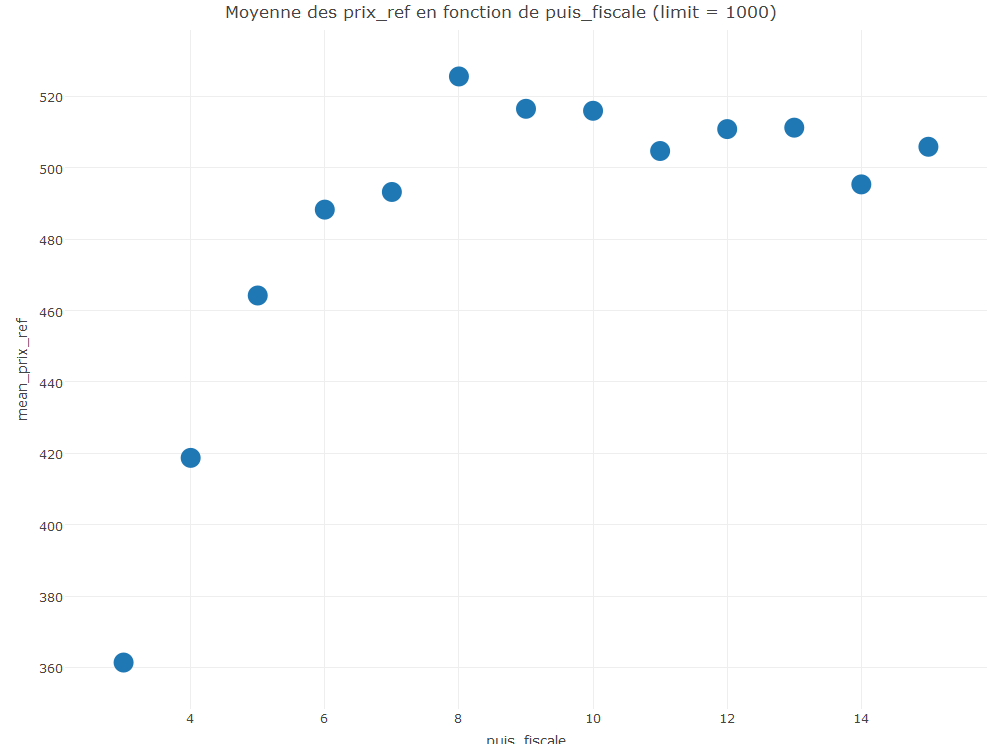
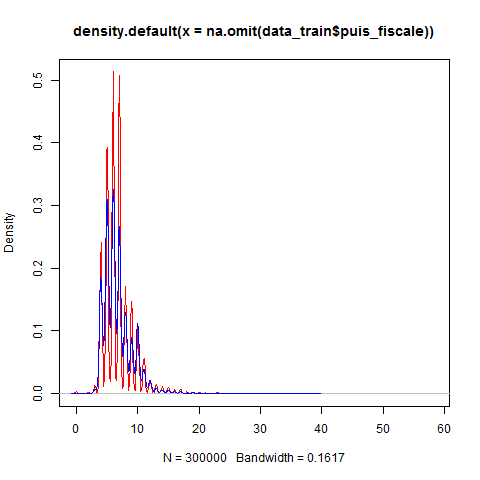
* Valeur abérante à remplacer par régression linéaire

## Marque :



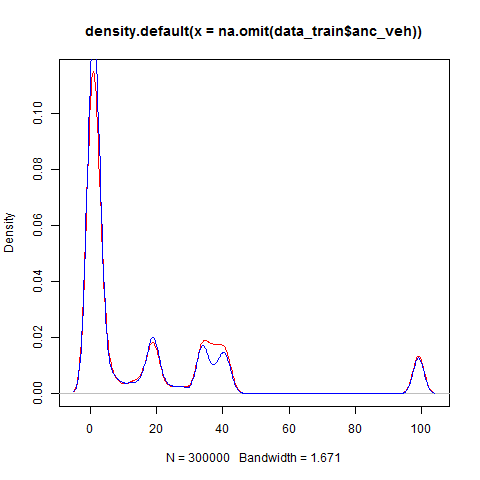
* Beaucoup de valeurs différentes (155 train vs74 test)
* 2 nouvelles valeurs dans test (GEELY/SOVAM)
* Var importante
* A traiter après ? soit cluster de marque + soit croisement avec gamme (var8)

## Puis\_fiscale :



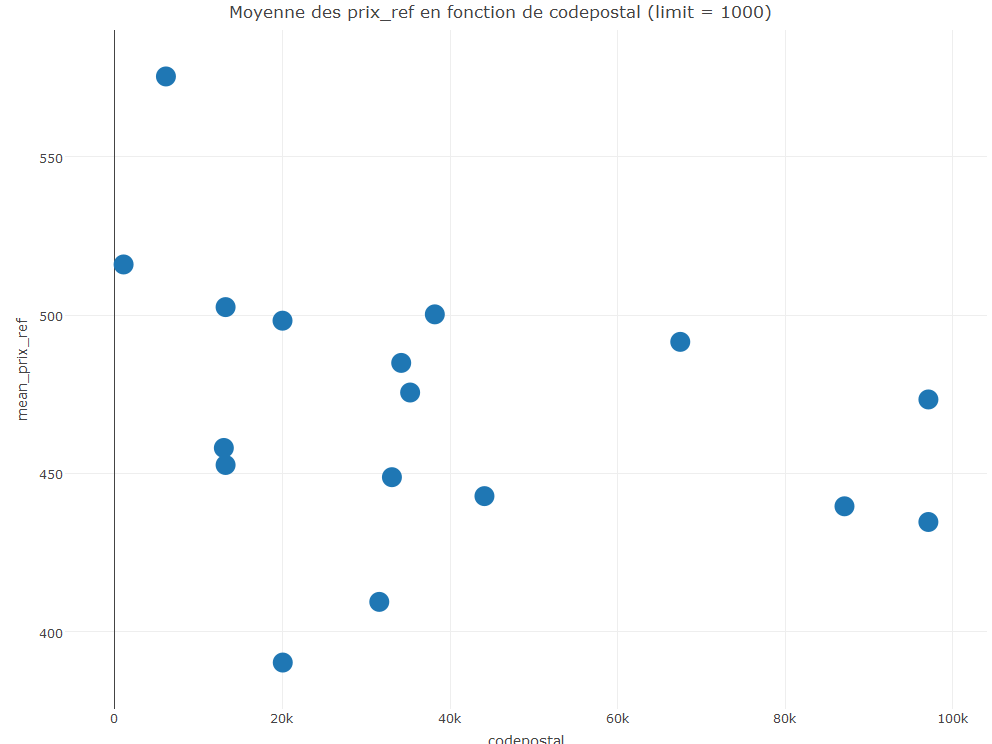
* Chevaux
* Plus éléve, plus chevaux, plus voiture puissante, plus prime élevé
* Effectif identique (train/test)
* En classe,
  + 2 classes
  + 6 classes

## Anc\_veh :



* Repartition identique
* Anciennté vehicule
* Var importante
* 2 types de clusters :
  + 4 classes basées sur distribution (ci-dessus)
  + 5 classes sur prix\_assurance

## Code postal :



Il y 23.696 codes postaux différents dans le train. Il semble que tous les département métropolitains soient représentés par contre.

On constate des valeurs numériques, 5 valeurs manquantes ‘NR’ et 3 fois la valeur « ARMEE ».

Dans le test seuls 2 départements sont présents pour 50 codes postaux: 01 et 97. 49 codes postaux pour 97 et 1 code postal pour 01.

Sur ces 50 codes postaux on en retrouve 3 dans le train : 01304 ; 97107 ; 97109.

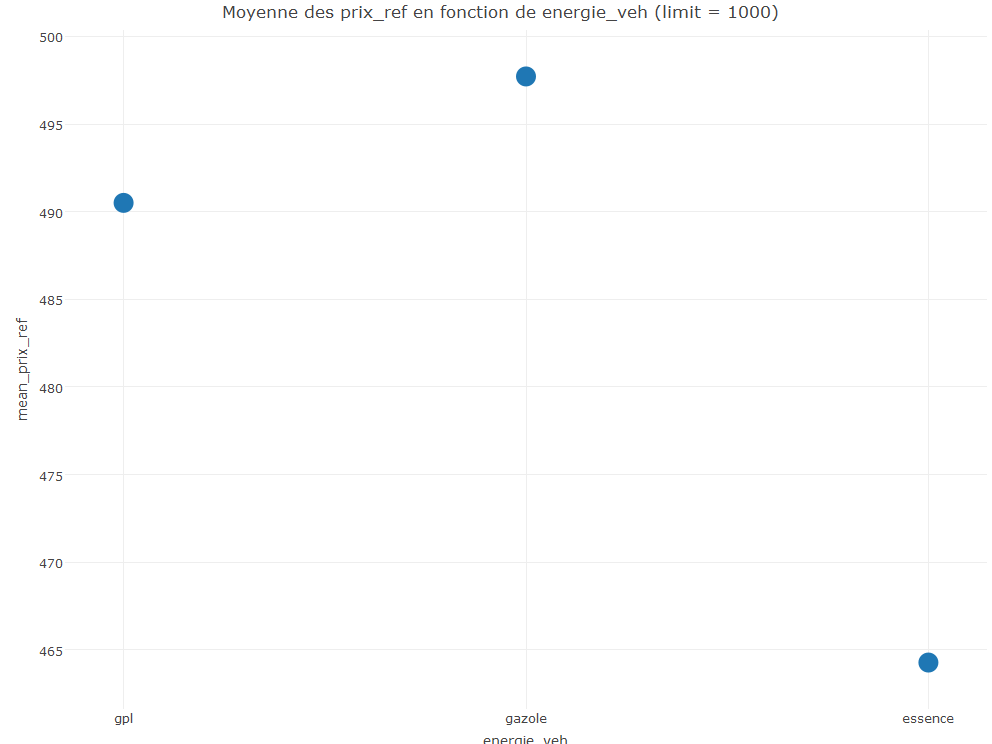
Affecter des poids par département permet d’obtenir de ‘légers’ meilleurs résultats. 15.91 -> 15.84

Affecter des poids par code postal dégrade le résultat : 15.91 -> 16.80

Hypothèse : Impact département (département indépendant des autres variables). But est de prédire prix\_assurance globale puis appliquer indice de département

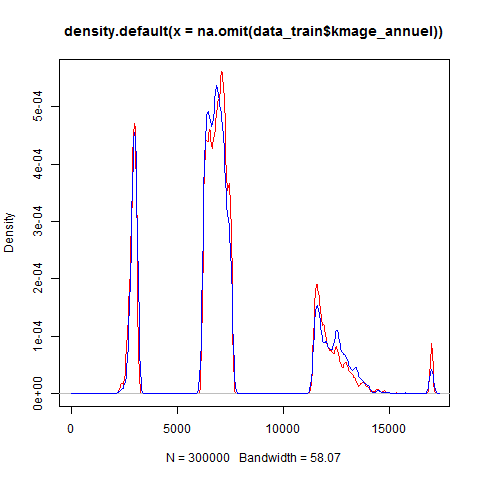
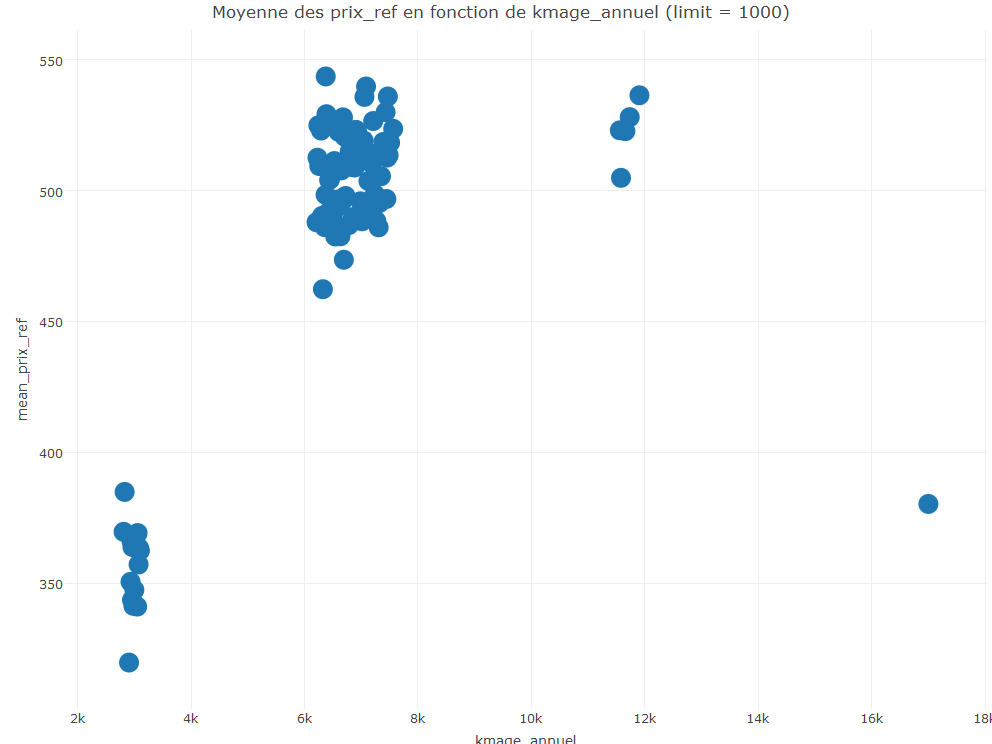
* Recodage avec 0
* Lissage par département (Valeur moyenne / valeur moyenne département)
* Var importante
* Transformation OUTPUT

## Energie\_veh



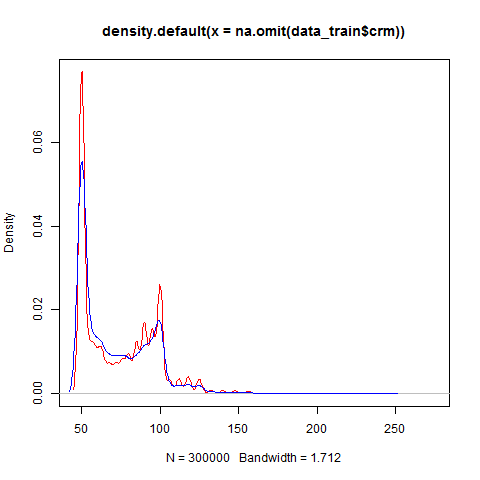
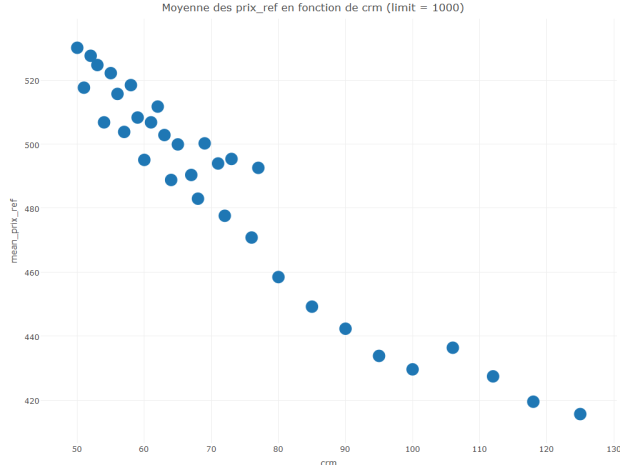
* Pas de modification
* Fusionner hybride-electricité

## Kmage\_annuel

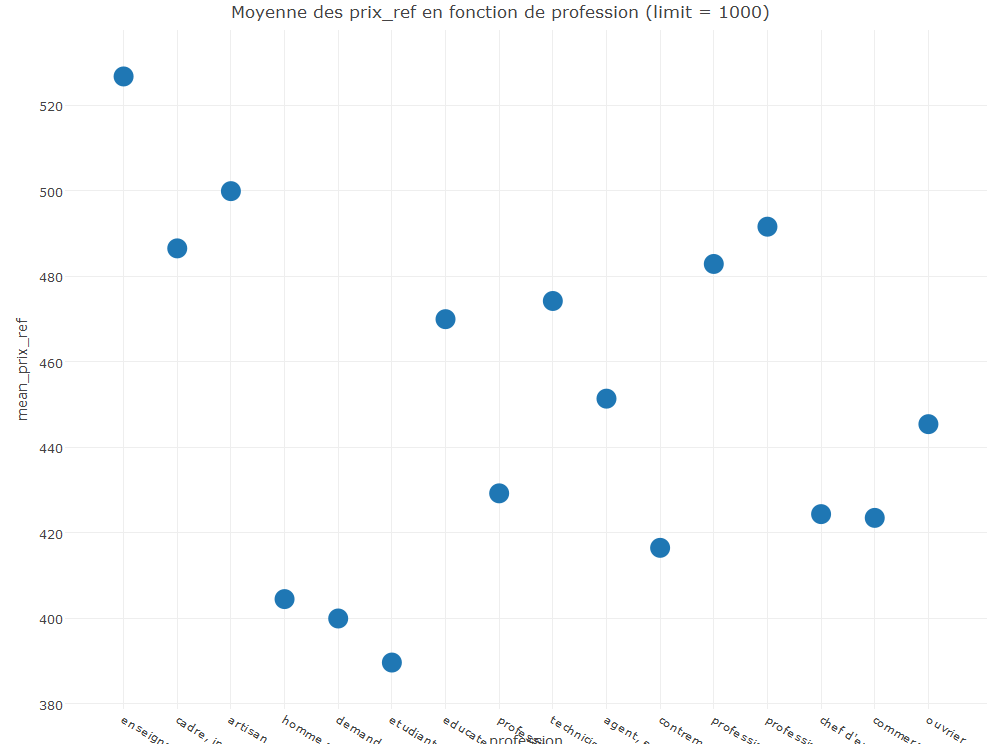
* Var importante
* Cluster :
  + Selon distribution, 4 groupes

## Crm

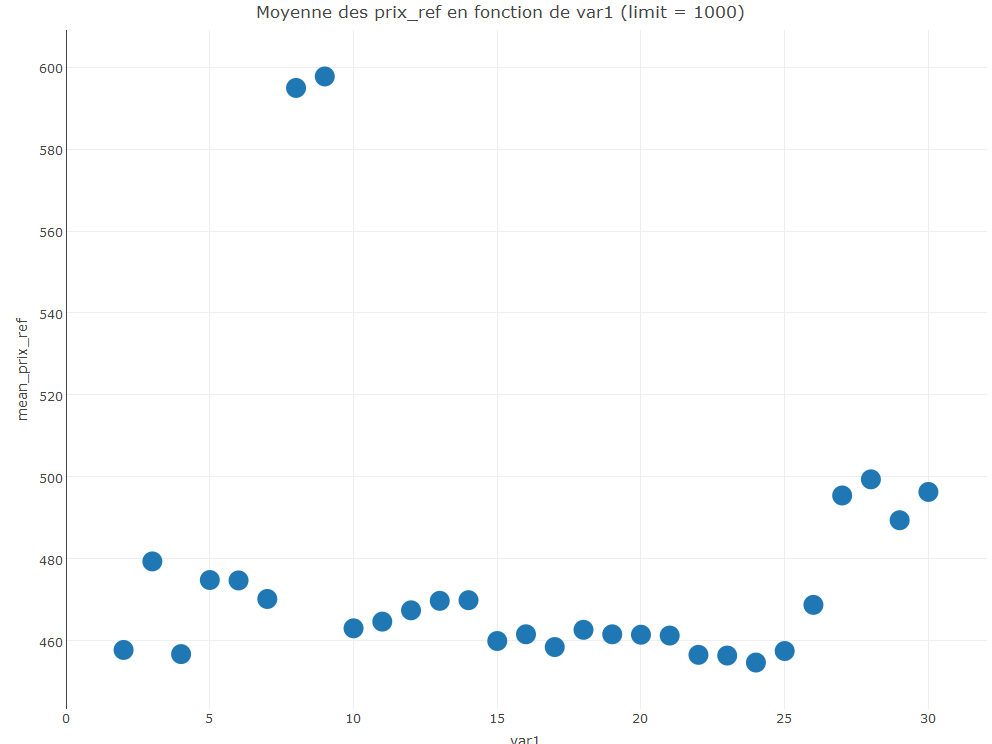
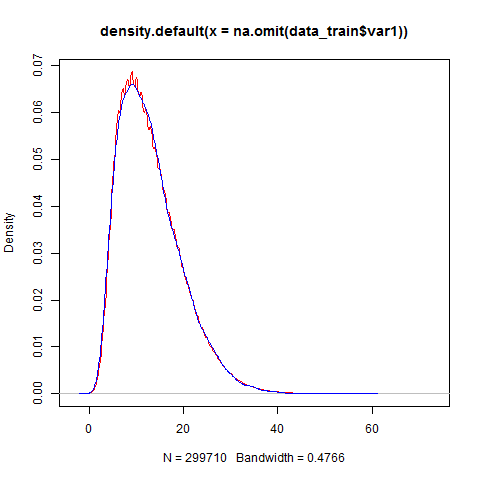
* Bonus/Malus
* Fonction à trouver (actuelle : prix\_ref\*CRM/100=prix\_tot\_ttc)
* Exclu des modèles
* Transformation OUTPUT

## Profession



* Hypothèse : pas d’impact dans prix assurance
* Apporte juste information (explicative)
* Cluster si besoin

## Var1

* Durée de permis

## Var2

## Var3

## Var4

## Var5

## Var6

## Var7

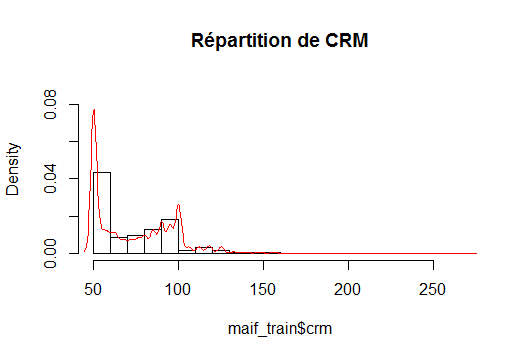
## Var8

## Var9

## Var10

## Variable CRM : bonus/malus

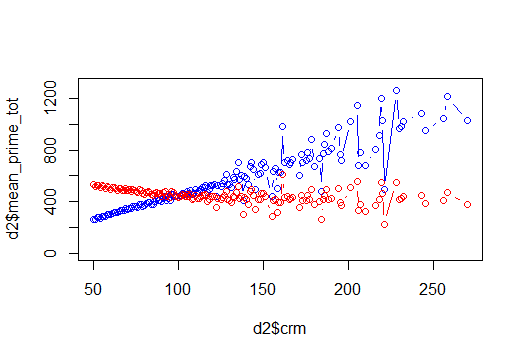
Il semble que la variable CRM corresponde au bonus/malus accordé pour l’assurance. Etant donnée la distribution ci-dessous, il semble que la valeur de CRM correspondent directement au nombre (\*100) par lequel la prime de base est multipliée pur obtenr le prix de l’assurance pour le conducteur.



CRM prend des valeurs allant de 50 à 270 mais très peu (moins de 0.01%) des valeurs sont supérieures à 195. Cela nous conduit donc à penser que CRM correspond au pourcentage du prix de base que va devoir payer le client.

On pense donc que la formule donnant le prix de l’assurance est :

On a donc comparé les prix de base et les prix finaux moyens par valeur de crm ; on obtient :



Les prix de la variable « prime\_tot\_ttc » sont en bleu, les prix de la variable calculée « prime\_brute » correspondant potentiellement au prix de base sont en rouge. Notre théorie semble se confirmer, la courbe pour le prix de base est beaucoup plus plate que la prime finale.   
Nous allons donc pour créer nos modèles utiliser la variable « prime\_brute » en output plutôt que la variable « prime\_tot\_ttc », variable que nous calculons avec la formule :

## Analyse de « codepostal »

# Pistes