SY19 - TP10

Hugo Martin, Louis Wahart, Zhifan Huang

May 13, 2023

1 Phonemes

Le jeu de données "phonemes" est tiré d'un ensemble de 4509 observations de 512 trames de discours, ayant été adaptées pour faire de la reconnaissance de la parole en log-périodogrammes composés de 256 variables. Parmi toutes ces données, seules 2250 ont été gardées pour entraînement. Ces données sont associées au son émis par 50 hommes lors de la prononciation d'un phonème, élément sonore du language parlé. Le problème posé est un problème de classification sur la variable y, qui représente le phonème prononcé parmi "aa", "ao", "dcl", "iy", "sh". Ce problème correspond donc à un problème de classification non binaire mais multiclasse.

1.1 Analyse exploratoire

Lorsqu'on regarde les données sans prendre en compte la classe associée aux observations, on se rend compte que la moyenne de toutes les variables se situe très proche de 0, et la variance très proche de 1: on peut donc supposer que les variables ont été centrées et réduites. En regardant les estimations des densités de chacune des variables, on voit que toutes les variables semblent être principalement uni-modales, et semblent presque suivre des gaussiennes.

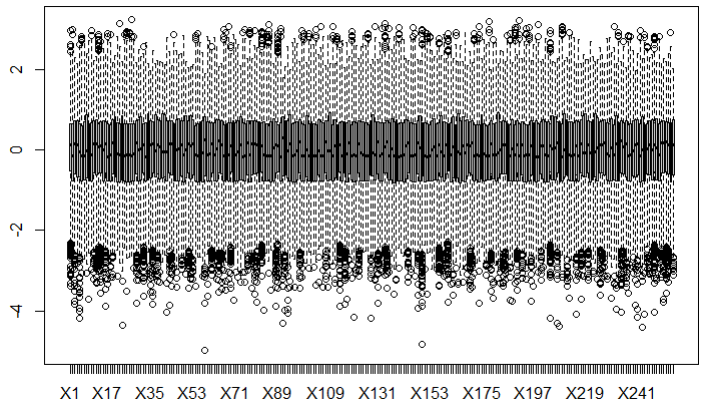


Figure 1: Boxplots de l'ensemble des variables

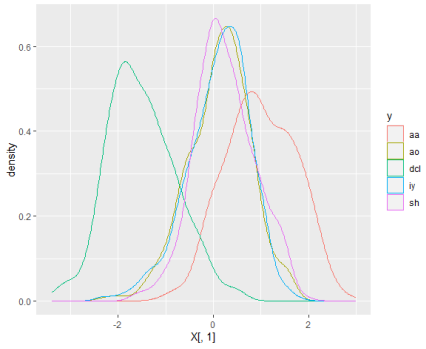


Figure 2: Estimation des densités pour X1

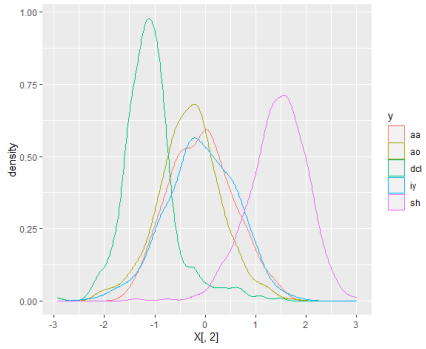


Figure 3: Estimation des densités pour X2

En ce qui concerne la variable de classe à prédire, nous avons remarqué que parmi les 5 phonèmes à prédire, deux d'entre eux se ressemblaient tout particulièrement, et que les 3 autres phonèmes étaient bien plus naturellement distinguables. En effet, les phonèmes "aa" et "ao" se ressemblent beaucoup lors de la prononciation orale, ce qui implique de fortes similitudes dans les données, tandis que les phonèmes "dcl", "iy", et "sh" sont très différents en termes de prononciation, et cela se ressent dans les données.

De plus, du fait de l'apparence de la distribution des variables, nous nous sommes demandé si il existait une simple relation linéaire entre plusieurs variables. Par conséquent, nous avons établi une matrice de corrélation, qui nous a permis d'observer que de nombreuses variables étaient très corrélées linéairement, et qu'il était possible de visualiser des blocs de variables corrélées pour les classes "aa" et "ao". Par exemple, on voit que parmi les premières variables, beaucoup semblent être très corrélées, ce qui a du sens car les deux phonèmes commencent par le même son, mais il est aussi intéressant de remarquer que les dernières variables ont tout de même un coefficient de corrélation élevé, ce qui laisse penser que le son "o" du phonème "ao" est un peu similaire au son "a" du phonème "aa". Nous avons utilisé le coefficient de corrélation de Pearson, qui permet justement de détecter la présence ou l'absence d'une relation linéaire entre des variables, ce qui était ce que nous voulions observer.

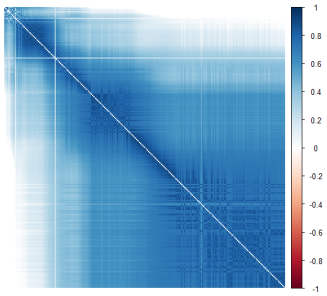


Figure 4: Graphe de corrélation pour toutes les classes (X1-X256)

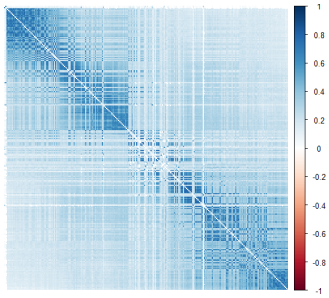


Figure 5: Graphe de corrélation pour les classes "aa" et "ao" (X1-X256)

1.2 Choix de modèle

Du fait de la proportion non négligeable de variables corrélées, nous avons gardé en tête que certaines des variables étaient possiblement redondantes, et qu'il serait peut-être judicieux de ne pas prendre l'ensemble des prédicteurs en compte pour notre modèle. En effet, notre réflexion était qu'un nombre de variables aussi important pouvait parfois rendre le modèle plus complexe qu'il avait besoin d'être, et que réduire le nombre de periodogrammes pris en compte dans le modèle pouvait nous permettre d'obtenir une meilleure précision.

De par la nature relativement uniforme des données et du grand nombre de prédicteurs, nous avons commencé par chercher sur des modèles qui profiteraient de la nature des données. Le fait que celles-ci semblent parfois suivre des pseudo-gaussiennes nous donnerait presque envie de tester un classifieur Bayésien naïf, mais le nombre de variables conséquent ne s'y prête évidemment pas. Ainsi, nous avons commencé par des modèles d'apprentissage basiques, ou basés sur une frontière de décision linéaire, comme un modèle des k plus proches voisins, ou une analyse discriminante linéaire (le nombre de variable fait que le jeu de données ne se prête pas à l'analyse discriminante quadratique). Puis, après des résultats corrects avec la LDA, nous avions tout de même envie de faire mieux, et nous avons alors essayé des modèles un peu plus complexes, comme les SVM ou les modèles basés sur des forêts aléatoires. Ces modèles fonctionnaient bien, mais obtenaient des taux d'erreurs assez similaires à ceux d'une LDA. Afin d'améliorer ces résultats, nous avons utilisé les hypothèses mentionnées précédemment concernant les corrélations entre les variables. Nous avons donc essayé de ne plus utiliser les variables "doublons", c'est-à-dire les variables parmi les couples de variables dont le coefficient de Pearson étaient supérieur à une certaine valeur, que nous avons déterminé par validation croisée, au même titre que le kernel du SVM, son hyperparamètre sigma (0.0168503129656732) ou encore la valeur *mtry* de la random forest.

1.3 Résultats et améliorations possibles

Comme on peut l'observer sur le tableau 1, les résultats du SVM et de la random forest sont légèrement meilleurs après avoir écarté les variables redondantes. Nous avons retenu le SVM de Weston-Watkins car il est réputé pour obtenir de très bons résultats dans des problèmes multi-classes, et correspondait également au type de SVM qui nous permettait d'avoir les meilleurs taux d'erreur parmi tous les types de SVM disponibles en R.

Table 1: Résultats récapitulatif des différentes méthodes utilisées, sur les données d'entraînement (taux d'erreurs obtenus avec une validation croisée)

Sans sélection de variables	Taux d'erreur
LDA	0.081
SVM (C support vector, cost C = 1, linear kernel)	0.105
SVM (Weston, Watkins native multi-class, radial basis kernel)	0.078
Random Forest (500 arbres, mtry=129)	0.078
Avec sélection de variables	
LDA	0.085
SVM (C support vector, cost C = 1, linear kernel)	0.103
SVM (Weston, Watkins native multi-class, radial basis kernel)	0.076
Random Forest (500 trees, mtry=104)	0.080

Cependant, ces résultats sont grandement impactés par la difficulté du classifieur à distinguer le phonème "aa" du phonème "ao". En effet, quand on sépare les deux problèmes (distinguer "aa" de "ao" et distinguer les 3 autres phonèmes), on se rend compte que la grande majorité des cas où le classifieur se trompe de prédiction correspond au problème de distinction entre les deux phonèmes mentionnés précédemment. Pour faire face à ce problème, plusieurs options s'offrent à nous. Nous pourrions combiner plusieurs classifieurs et choisir la classe en fonction de la probabilité moyenne, calculée en combinant les probabilités des différents modèles, ou bien choisir uniquement la deuxième moitié des variables dans le cas des deux phonèmes difficiles à séparer, étant donné que les différences ont toutes les raisons de se trouver à la fin du son. Nous pourrions également séparer le problème en deux étapes: la première consisterait à entraîner notre modèle le plus efficace sur toutes les données, en prenant en compte toutes les classes. Ensuite, il s'agit de trouver un modèle permettant de très bien séparer les deux phonèmes que nous avons du mal à distinguer et de l'entraîner sur les données dont la prédiction était soit "aa" soit "ao", puis de faire une nouvelle prédiction de ces données, cette fois plus spécifiquement. Cependant, la mise en place de cette solution et des autres tentatives n'ont pas eu les résultats escomptés, et nous n'avons jamais vraiment réussi à déroger aux 20 pourcents environ d'erreur de prédiction pour les deux premiers phonèmes. Une de nos hypothèses à ce sujet est que si la mise en place de ces dispositifs ne nous permet pas de nous écarter d'un certain taux d'erreur non-négligeable (environ 20 pourcents), alors il nous faudrait peut-être changer de base, et essayer d'effectuer une ACP. En effet, peut-être qu'en effectuant un changement de base, on pourrait faire ressortir les features les plus importantes pour faire une meilleure distinction entre les classes, en se libérant du facteur de corrélation entre les variables.

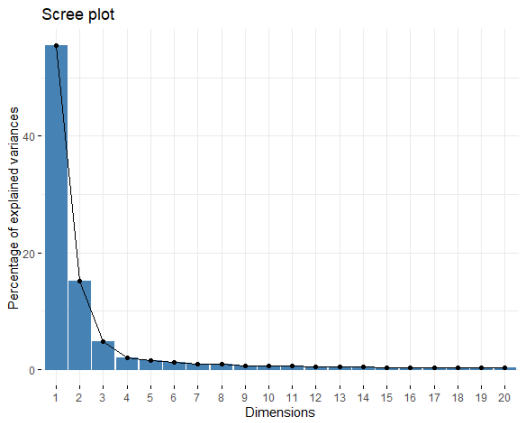


Figure 6: Pourcentages de variance expliquée par les axes tirés de l'analyse en composantes principales

Après avoir effectué une PCA sur les données, on observe le fameux "coude" sur la figure 6 lorsqu'on affiche la variance expliquée par chacun des axes, ce qui confirme l'utilité d'une PCA, et nous permet ainsi de n'utiliser qu'une petite partie des axes. En utilisant les 40 premiers axes, nous avons pu obtenir des résultats légèrement meilleurs, mais sans changement radical (avec un SVM de Weston Watkins à noyau radial, par validation croisée: 0.072).

float

2 Robotics

The Robotics dataset contains data related to the kinematics of a robot arm. It has eight features and one response 'y'. And, y is a continuous number, so the learning task is a regression problem. Because these predictors describe the movement of a robot arm, they may be related to six degrees of freedom. So the non-leaner model may be better than the leaner model. In addition, the eight variables don't have names, so the model does not need to have a good comprehensibility. But, expectation maximization algorithm can't be used, because we don't have any former knowledge based on these feature

2.1 Analyzing Data

In the boxplot of variables, the eight features have almost the same range, they may have the same distribution. Furthermore, their mean is close to zero, and their variance is about 1. So they have been normalized. Also their covariances are around 0.01, so they are not linearly correlated. The Figure 7 shows the distribution of X2, whose middle part is nearly uniform, with nothing special.

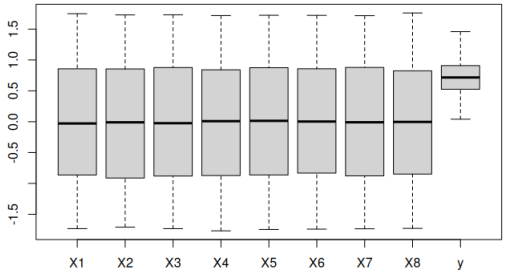


Figure 7: Boxplot of variables

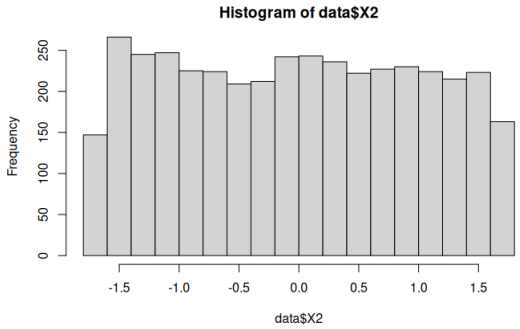


Figure 8: Hist of variables X2

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Standard deviation	1.0331	1.0192	1.0130	1.0116	0.9948	0.9876	0.9795	0.9741
Proportion of Variance	0.1329	0.1294	0.1278	0.1275	0.1232	0.1215	0.1195	0.1182
Cumulative Proportion	0.1329	0.2623	0.3901	0.5176	0.6408	0.7623	0.8818	1.0000

This table shows the importance of robotic's features, as generated by the command prcomp. Each component has a similar variance and almost has its own unique role. So Dimensionality reduction becomes a difficult task. So methods for decreasing dimensions like subset selection and shrinkage are not attractive anymore. And consider it's density $4000^{1/8} = 2.82$. For any point, there are not enough adjacent points to predict the response y. k-means will lead to the curse of dimensionality.

2.2 Linear Model

Because of the simplicity of linear regression, we can try it out and deepen our understanding of the data. However, as previously guessed, the linear model may not be suitable for this dataset. According to the summary of the linear model, each feature's p-value is extremely little, which corresponds to the previous component analysis. However, the adjusted R-squared is 0.4015. This means the model failed to predict the response y. 50% variance of y is not explained by this model. The MSE on test data is 0.04 The photo shows the Q-Q plot of the residual.

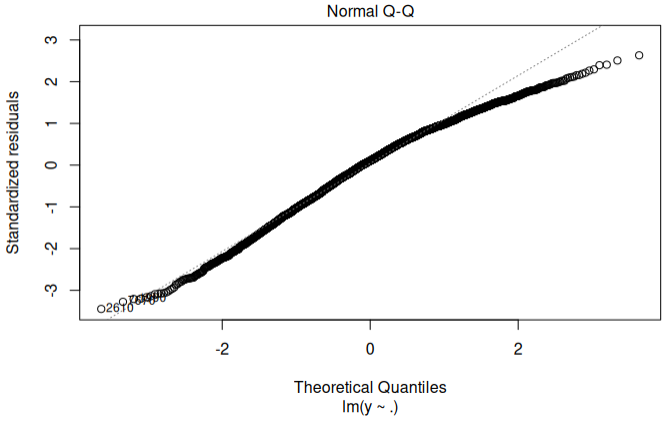


Figure 9: Linear regression residual˙qqplot

When the residuals are greater than zero, they do not satisfy the normal distribution, which means the assumption of epsilon failed. Since the value of y is greater than 0, I also tried to take the log of y, but it did not improve the residuals' problem.

I then tried using nature spline to see if the degree of each feature would improve the results. By using the following formula. But that also failed.

```
y~ns(X1,df=3)+ns(X2,df=3)+ns(X3,df=3)+ns(X4,df=3)+ns(X5,df=3)+ns(X6,df=3)+ns(X7,df=3)+ns(X8,df=3)
```

Finally, I correlate multiple variables by the formula y X1*X2*X3*X4*X5*X6*X7*X8. It is exciting to see that the results have improved significantly, the R-squared increase to 0.7, and the test MSE drop to 0.02. So these variables are not independent of each other.

2.3 Non-linear Model

2.3.1 Random Forest

Since there are only 8 variables, the value of mtry for the random forest is 3 regardless of whether the default n/3 or use log2(n). And the test MSE is not special, 0.021. By increasing the ntree parameter, the MSE little optimized from 0.021 to 0.020. And the result of bagging is similar to the random forest.

2.3.2 Support Vector Machine

As said before, linear models are not very good, so kernel tricks are important to nonlinear transform the input feature space to a new feature space.

rbfdot	polydot	vanilladot	tanhdot	laplacedot	besseldot	anovadot
0.01271	0.04295	0.04295	662.1834	0.0156	0.2601	22.2761'

The table above shows the cross-validation result of different kernels. The rbfdot and laplacedot have the best result. Their MSEs are already better than previous models. Besides, polydot use polynomial of features, it's MSE closed to the spline models.

We then focused on their different parameters by using cross-validation.

The grid of cross-validation about cost and sigma with kernel laplacedot

C sigma	0.03125	0.0625	0.125	0.25	0.5	1
32	0.008338947	0.008306725	0.008107607	0.008436020	0.008921340	0.01079038
64	0.008339537	0.008212678	0.008330160	0.008416510	0.008672814	0.01099427
128	0.008256747	0.008220027	0.008400844	0.008296268	0.008876666	0.01081118
256	0.008145284	0.008213203	0.008224357	0.008380859	0.008837824	0.01105220
512	0.008309967	0.008223594	0.008313709	0.008319953	0.008829467	0.01085927
1024	0.008257692	0.008198213	0.008409451	0.008416197	0.008759666	0.01093048

The grid of cross-validation about cost and sigma with kernel rbfdot

C sigma	0.03125	0.0625	0.125	0.25	0.5	1	2
0.5	0.02685705	0.019234443	0.011482773	0.008134370	0.010568028	0.02630474	0.05855071
1	0.02476976	0.016385046	0.008990384	0.007034462	0.009253777	0.02068916	0.05256697
2	0.02198867	0.013724261	0.007574106	0.006673018	0.008719502	0.01997557	0.04971728
4	0.01941255	0.011470439	0.006813427	0.006739043	0.009127632	0.02004326	0.04973362
8	0.01739689	0.009400735	0.006567940	0.007250834	0.008966554	0.02020160	0.04970631
16	0.01569961	0.008281557	0.006474320	0.007836754	0.008923694	0.01992549	0.04976910
32	0.01319922	0.007324950	0.006771780	0.007980618	0.009191220	0.02011846	0.04951920
64	0.01098693	0.007087883	0.007666046	0.008211932	0.009254396	0.02005148	0.04946180

These two tables show how the two kernels behave under different parameters. The parameter interval is exponential in two to cover more ranges. For laplacedot, its performance verify on the cost and sigma with no evidence of tendency. When the sigma is smaller than 0.25, the MSE seems to change randomly based on the cost. For rbfdot, it has better performance than laplacedot. Furthermore, it has a clear minimum MSE interval. MSE is minimized when sigma around 0.25, cost around 16.

Because epsilon defines a margin of tolerance where no penalty is given to errors and error is determined concerning cost. So I do another grid search with epsilon and cost.

The grid of cross-validation about cost and epsilon with kernel rbfdot

C epsilon	0.0125	0.025	0.05	0.1	0.2	0.4
6	0.006668170	0.006806161	0.006589806	0.006670819	0.006711102	0.008297290
8	0.006685697	0.006597636	0.006554867	0.006409632	0.006748865	0.008125630
10	0.006718945	0.006549523	0.006505093	0.006609048	0.006706116	0.007997654
12	0.006757601	0.006577276	0.006493934	0.006553620	0.006755135	0.007979025
14	0.007013213	0.006752403	0.006716760	0.006607136	0.006664013	0.008067838
16	0.006782465	0.006810728	0.006617948	0.006580512	0.006762107	0.008226855

The result indicates that epsilon = 0.1 is the best choice. Finally, the following command is used to train SVM model.

```
ksvm(y~., data=data, kernel='rbfdot', C=8, epsilon=0.1, kpar=list(sigma =0.125))
```

2.3.3 Neural Network

The cross-validation of hidden units number in nnet

29	36	43	50	57	64
0.006773181	0.006802221	0.006909978	0.006457816	0.006649864	0.006742320
71	78	85	92	99	
0.006483055	0.006189515	0.006319409	0.005934737	0.005896900	

The neural network has a good performance in predicting the response. It's performance increases with the number of hidden neurons.

2.4 Conclusion

For this dataset, the nonlinear model is better than the linear model. However, by adjusting the linear regression formula, like adding cross terms, the model can be improved, even comparable to the random forest. SVM and neural networks perform the best, probably because the features record the space coordinates of the robot arm. While these two models can extract information like trajectory shapes (circles, ellipses) from them, so they perform better.

3 Communities

Ce jeu de données contient 1000 observations de 122 variables socio-économiques et policières recueillies dans autant de communautés aux États-Unis dans les années 90. Nous devons construire un modèle permettant de prédire le nombre de crimes violent pour 100,000 personnes d'une part et d'autre part déterminer quelles variables recueillies influent le plus sur cette valeur.

3.1 Prétraitement

On peut tout d'abord noter que l'on disposait également de 5 variables dites non-prédictives. Parmi ces variables on a le numéro de l'État, ce qui pourrait avoir une valeur prédictive, puisque certains États pourraient avoir un taux de crimes violents plus importants, mais cela n'apporterait que peu d'informations par rapport à l'analyse socio-économique des variables associées à ce taux. On ne prendra donc pas en compte cette variable. Il n'est ici pas nécessaire de standardiser les variables car d'après le descriptif du jeu de données elles ont toutes déjà été normalisées et sont donc sur la même échelle, comme on peut le voir sur la figure 10. Par ailleurs on remarque que deux variables, "PolicePerPop" et "LemasSwornFT" ont une corrélation de 1, elles ont toujours la même valeur. On supprime donc la deuxième variable du jeu de données. Nous séparons le jeu de données en un ensemble d'entraînement et un ensemble de test selon un découpage 80%/20%.

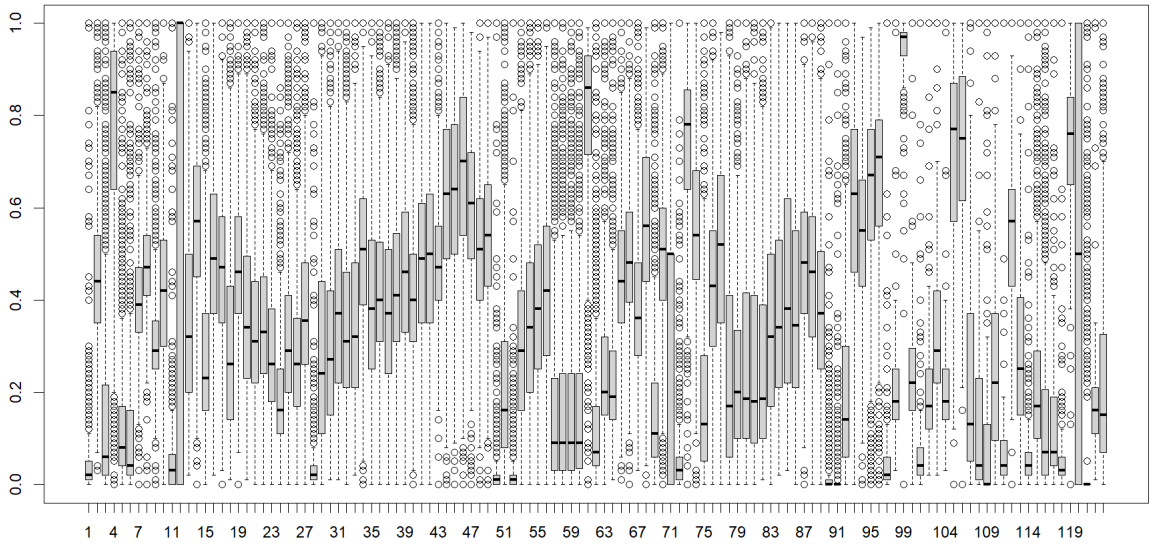


Figure 10: Boxplots de l'ensemble des variables

3.2 Gestion et analyse des données manquantes

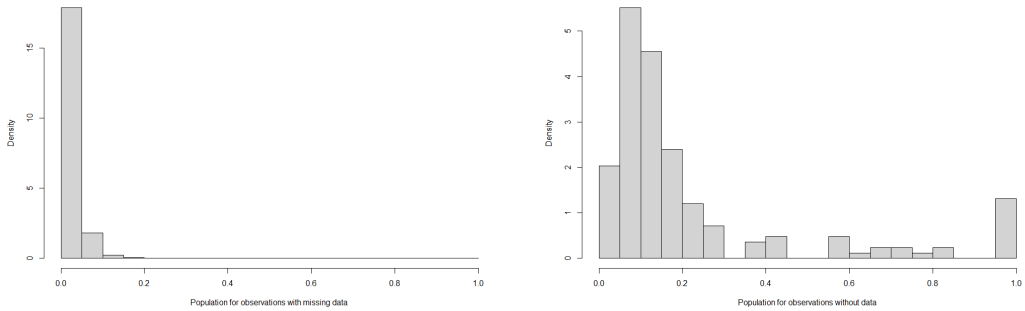
Il manque la donnée "OtherPerCap" pour une des communautés sans qu'il n'y ait de raison donnée, nous supposons donc que cette absence est complètement aléatoire. Nous décidons ainsi de remplacer cette donnée par une valeur représentative de cette variable. Nous réalisons un test de normalité de Shapiro-Wilk pour déterminer si

la moyenne empirique est une statistique représentative de la distribution pour imputer cette donnée. Nous trouvons une p-value inférieure à $2.2 * 10^{-16}$, on peut rejeter l'hypothèse nulle de normalité avec une grande confiance. On décide donc d'utiliser la médiane plutôt que la moyenne pour cette variable.

Le reste des données manquantes touche les variables liées à l'état des forces de police dans les communautés étudiées. Pour ces variables, nous envisageons trois approches différentes :

1. Suppression des variables aux données incomplètes
2. Suppression des communautés aux données incomplètes
3. Imputation des données manquantes

La première approche a un problème évident : on ne pourra pas déterminer l'impact des variables supprimées sur la variable cible. La seconde approche pose deux problèmes. D'abord les observations concernées par les données manquantes sont très nombreuses : 833 sur 1000. Ensuite, les données ne sont pas manquantes de manière aléatoire. En effet d'après la description du jeu de données, les données liées aux forces de police sont issues de l'étude LEMAS qui ne concernait que les départements de police comptant plus de 100 officiers, ainsi qu'une sélection aléatoire de plus petits départements. Les données manquantes correspondent donc toutes à des départements de police comptant moins de 100 officiers (petites communautés, communautés "tranquilles", communautés en manque de moyen, etc), ce qui peut introduire un biais dans l'analyse et conduire à de moins bons résultats sur les données tests, où l'on devra appliquer le modèle à des communautés ayant toute taille de départements de police. De plus, même en supprimant les lignes concernées il sera nécessaire de trouver une méthode d'imputation pour les variables manquantes des données de test. La dernière méthode envisagée est l'imputation des données manquantes. Comme précédemment on pourrait remplacer les données manquantes par la moyenne ou la médiane des données. Ici cela ne nous semble pas pertinent puisque comme expliqué précédemment les données manquantes ne sont pas complètement aléatoires mais liées aux nombre de policiers dans le département de police de la communauté. Ainsi la distribution de certaines variables au sein des observations incomplètes est parfois très différente de celles observée pour les données complètes, comme on peut le voir sur la figure 11 pour l'attribut "population" par exemple. On peut supposer que cela soit également le cas pour certaines des données manquantes. En faisant la médiane ou la moyenne, on inclurait donc dans le calcul des forces de police de petits départements des données de départements bien plus grands. Nous pouvons tenter de construire un modèle de prédiction pour chaque attribut en fonction des autres attributs (complets) présents dans le jeu de données.



(a) Données manquantes

(b) Données complètes

Figure 11: Distribution de l'attribut "population" parmi les données manquantes et les données complètes

3.3 Imputation des données de police manquantes

Pour chaque variables manquantes nous construisons un modèle de prédiction à partir des autres features. Nous décidons de nous restreindre à des modèles linéaires obtenus par sélection itérative des variables en "forward" basée sur le R^2, puis par une comparaison des meilleurs modèles à n variables par validation croisée ($k = 5$ répétée 3 fois).

3.4 Modèles de prédiction

Nous voulons obtenir un modèle interprétable, nous pouvons donc par exemple construire un modèle linéaire pour prédire la variable cible. Le coefficient associé à une variable X_i représentera l'augmentation répercutée sur la variable cible si l'on augmente la variable X_i d'une unité. On apprend tout d'abord un modèle linéaire utilisant toutes les variables avec une régularisation sur les paramètres de régularisation élastique α et λ, en incluant le cas d'une abscence de régularisation ($\alpha = 0$ et $\lambda = 0$). Nous entraînons trois modèles de ce type sur les trois jeu de données décrits précédemment. On teste ensuite un modèle obtenu de la même manière que les modèles de la section précédente (sélection itérative). Enfin, on teste un modèle gardant les 50 premières variables sélectionnées itérativement tout en appliquant une régularisation élastique. L'erreur quadratique moyenne sur le jeu de test (MSE) et le R^2 pour chacun de ces modèles sont rapportés en table **??**.

Nous entraînons également des modèles de forêt aléatoire sur chacun des jeux de données, qui ne seront pas interprétables mais qui pourront peut-être fournir des prédictions plus précises. Pour ce faire, on effectue dans le cas du jeu de donnée imputé une validation croisée ($k = 10$, 5 répétitions) avec différentes valeurs de *mtry*, le nombre de variables considérées à chaque division. On fait une première validation croisée avec des valeurs assez éloignée les unes des autres sur l'espace de la variable : 3, 10, 20, 50, 80, 100. On obtient les meilleurs résultats pour les valeurs 10 (0.01922) et 20 (0.01923). On effectue ensuite une seconde validation croisée d'affinement avec les valeurs de 8 à 30 par pas de 2. On obtient le meilleur résultat avec la valeur 24 (0.01789). On obtient une valeur de 0.01948 sur le jeu de données de test. On répète les mêmes opérations pour les deux autres jeux de données, mais on obtient de moins bons résultats.

3.5 Analyse de l'influence des variables sur le taux de crimes violents

Avant de s'intéresser aux variables les plus influentes/importantes des modèles précédents, il convient de rappeler que d'une part les modèles linéaires ne prennent pas en compte l'interaction des variables entre elles, et d'autre part ces modèles n'établissent pas des liens de causalité mais des associations entre les variables. Ainsi il est parfois possible de prédire efficacement une variable Y avec une variable X_1 n'ayant pas de lien causal (dans un sens ou l'autre) Y mais ayant un "parent commun" X_2 ayant un effet causal sur X_1 et Y. On emploiera donc les termes "effet", "importance", "influence" dans cette section dans le sens d'intensité de l'association apprise par les modèles, et non d'une quelconque causalité supposée.

Les modèles linéaires développés précédemment sont tous interprétables, on peut savoir dans quelle mesure chaque variable a participé à la prédiction. Les coefficients étant sur la même échelle de 0 à 1, on peut calculer la moyenne et l'écart-type des coefficients obtenus avec chaque modèle développé pour les jeux de données 1 et 3 (présentant chacun toutes les variables) pour obtenir une valeur permettant de comparer l'influence des coefficients entre eux. On trouve les valeurs présentées en figure 12.

On remarque que l'ensemble des modèles sont globalement unanimes en ce qui concerne le sens de l'influence des variables les plus importantes. Par exemple pour tout les modèles "PctKids2Par" (pourcentage d'enfants vivant en famille avec deux parents) semble avoir un effet négatif pour le taux de crimes violents, ce qui paraît intuitif. On note cependant pour ce coefficient et pour nombre d'autres une variance assez importante entre les modèles pour le jeu de données 2, par exemple pour "PctTeens2Par" qui passe de 0 à 0.98. Cela est probablement du au manque de données disponibles pour ce jeu de données, puisque les coefficients des modèles développés sur les autres jeux de données présentent des variances moins importantes.

Les variables liées aux forces de police, présentes dans les jeux de données 2 et 3, semblent avoir chacune un peu d'influence sur la variable cible.

Parmi les variables qui semblent être les plus importantes selon ces modèles, on peut citer "PctKids2Par", "racepctblack", "NumStreet", ainsi que des variables mesurant le capital économique des communautés (données liées au loyer, au revenu, aux emplois majoritaires, etc suivant les modèles), toujours présentes dans les dix variables les plus importantes de chaque modèle. Les variables du jeu de données présentent un grand nombre de corrélations, il est donc possible que chaque modèle choisisse plus ou moins aléatoirement une variable représentative de la situation économique, et extraient le nécessaire d'information économique de celle-ci sans avoir besoin du reste des informations du même type.

Le problème des modèles régularisés est que l'on a pas de p-value associée à chaque coefficient, on a donc pas d'assurance que le coefficient soit différent de 0 de manière statistiquement significative. On a cependant accès aux p-values des coefficients des variables des modèles linéaires aux variables sélectionnées itérativement, que l'on peut lire figure 13.

On retrouve dans les trois modèles les coefficients statistiquement significativement différents de 0 (à 95%) associés aux variables suivantes : "NumStreet" (nombre de sans-abris, positivement), "MedOwnCostPctIncNoMtg"

	coeffs elastic net	coeffs iterative selection	coeffs it. sel. + elastic net	mean	std
PctKids2Par	-0.300650231	-0.31819991	-0.355955034	-0.324935057	1.597354e-03
racepctblack	0.206645678	0.36156693	0.109062051	0.225758219	3.242729e-02
PctPersDenseHous	0.136396356	0.18481573	0.137016807	0.152742966	1.543186e-03
NumStreet	0.098112767	0.15225736	0.086604655	0.112324926	2.458117e-03
HousVacant	0.069610608	0.12482550	0.118873213	0.104436442	1.836973e-03
RentLowQ	0.000000000	-0.28867777	0.000000000	-0.096225923	5.555657e-02
MalePctDivorce	0.093407642	0.12395623	0.061159047	0.092840972	1.972225e-03
MedRent	0.000000000	0.23228509	0.000000000	0.077428362	3.597091e-02
MedOwnCostPctIncNoMtg	-0.072768852	-0.13229918	-0.014626442	-0.073231493	6.923758e-03
PctIlleg	0.112880714	0.00000000	0.102497724	0.071792813	7.785215e-03
PctNotSpeakEnglWell	0.000000000	-0.20715887	0.000000000	-0.069052957	2.860986e-02
PctRecImmig8	0.000000000	0.18682710	0.000000000	0.062275701	2.326958e-02
PctHousOccup	-0.060215161	-0.10083788	-0.024707349	-0.061920129	2.902289e-03
PctNotHSGrad	0.016059469	0.13504905	0.000000000	0.050369506	1.088489e-02
PctPopUnderPov	0.000000000	-0.14348586	0.000000000	-0.047828620	1.372546e-02
PctWorkMomYoungKids	-0.038693969	-0.09710473	-0.005949897	-0.047249531	4.264398e-03
racePctHisp	0.010484915	0.12586035	0.000000000	0.045448421	9.754084e-03
PctImmigRec10	0.000000000	-0.11000422	0.000000000	-0.036668073	8.067286e-03
MedRentPctHousInc	0.011090623	0.08317142	0.000000000	0.031420680	4.078709e-03
racePctWhite	-0.029366725	0.04425087	-0.015905235	-0.030340362	1.127485e-02
PctUrban	0.025559589	0.04246904	0.018015755	0.028681462	3.136008e-04
PctVacMore6Mos	0.000000000	-0.07420500	0.000000000	-0.024735001	3.670922e-03
pctWFarmSelf	0.000000000	0.06938543	0.000000000	0.023128476	3.209558e-03
numbUrban	0.046132934	0.00000000	0.019943648	0.022025527	1.070625e-03
MedYrHousBuilt	0.000000000	-0.05651399	0.000000000	-0.018837996	2.129221e-03
PctForeignBorn	0.045646047	0.00000000	0.002638898	0.016094982	1.313380e-03
NumInShelters	0.036033753	0.00000000	0.000000000	0.012011251	8.656209e-04
agePct12t29	0.000000000	0.03074501	0.000000000	0.010248336	6.301703e-04
PctImmigRecent	-0.022739582	0.00000000	0.000000000	-0.007579861	3.447257e-04
pctWSocSec	0.022195962	0.00000000	0.000000000	0.007398654	3.284405e-04
PersPerRentOccHous	0.009036660	0.00000000	0.000000000	0.003012220	5.444082e-05
AsianPerCap	0.006722474	0.00000000	0.000000000	0.002240825	3.012777e-05
PctSameCity85	0.003687108	0.00000000	0.000000000	0.001229036	9.063179e-06
MedNumBR	-0.003032770	0.00000000	0.000000000	-0.001010923	6.131797e-06

(a) Jeu de données 1

	coeffs elastic net	coeffs iterative selection	coeffs it. sel. + elastic net	mean	std
PctKids2Par	-0.152326161	-1.26607575	-0.602138600	-0.6735135049	6.278606e-01
PctTeen2Par	0.000000000	0.98789070	0.000000000	0.3293302324	6.507504e-01
PctNotHSGrad	0.000000000	-0.94773595	0.000000000	-0.3159119841	5.988023e-01
PctForeignBorn	0.000000000	0.63568273	0.020730618	0.2188044488	2.608961e-01
PctEmplManu	-0.063903959	-0.40617838	-0.154375173	-0.2081325043	6.291389e-02
PctWOFullPlumb	0.000000000	-0.44622113	-0.006915400	-0.1510455093	1.307169e-01
PctHousLess3BR	0.077600408	0.30247216	0.068753227	0.1496086001	3.509004e-02
PctUnemployed	0.028643228	0.31017182	0.103687803	0.1475009512	4.250856e-02
racepctblack	0.046687242	0.32632776	0.062503926	0.1451663085	4.935461e-02
PctLess9thGrade	0.000000000	0.40646062	0.000000000	0.1354868729	1.101402e-01
perCapInc	0.000000000	-0.33994178	-0.009430586	-0.1164574556	7.496233e-02
NumStreet	0.039877973	0.10696164	0.142235077	0.1161915635	1.046153e-03
PctRecentImmig	0.000000000	-0.30224535	0.000000000	-0.1007484516	6.090150e-02
LemasPctOfficDrugUn	-0.001991436	-0.20985297	-0.077430818	-0.0964250742	2.214438e-02
pctWPubAsst	0.012066432	0.26834181	0.000000000	0.0934694152	4.594333e-02
PctBornSameState	0.000000000	-0.21366709	-0.060853076	-0.0915067229	2.423628e-02
PctHousNoPhone	0.000000000	0.25633547	0.000000000	0.0854451570	4.380525e-02
LandArea	0.000000000	0.15092750	0.095066088	0.0819978628	1.164572e-02
racePctWhite	-0.107905398	-0.02748883	-0.106908167	-0.0807674639	4.258416e-03
PctPersDenseHous	0.026916422	-0.27458081	0.009902342	-0.0792540145	5.737357e-02
PctOccupManu	0.000000000	0.23030842	0.000000000	0.0767694721	3.536131e-02
PctPolicAsian	0.044785769	0.05510476	0.097813474	0.0659013334	1.580818e-03
LemasGangUnitDeploy	0.058950077	0.04274891	0.084059797	0.0619195957	8.665216e-04
NumKindsDrugsSeiz	0.000000000	0.15551072	0.029145410	0.0615520448	1.366708e-02
PolicBudgPerPop	0.000000000	0.16668428	0.000000000	0.0555614260	1.852243e-02
LemasPctPolicOnPatr	0.000000000	0.13833676	0.001138082	0.0467391405	1.272391e-02
PctFam2Par	-0.108754686	0.00000000	0.000000000	-0.0362515621	7.885055e-03
PctHousOccup	0.000000000	0.10783690	0.000000000	0.0359456317	7.752531e-03
PctYoungKids2Par	-0.076696093	0.00000000	0.000000000	-0.0255633644	3.921527e-03
PctIlleg	0.066091768	0.00000000	0.000000000	0.0220305892	2.912081e-03
PctBSorMore	0.000000000	0.00000000	-0.058835138	-0.0196117127	2.307716e-03
PctImmigRecent	0.000000000	0.00000000	-0.052813735	-0.0176045783	1.859527e-03
pctWInvInc	-0.043087657	0.00000000	0.000000000	-0.0143625524	1.237697e-03
PolicCars	0.039503154	0.00000000	0.000000000	0.0131677179	1.040333e-03
TotalPctDiv	0.032822805	0.00000000	0.000000000	0.0109409348	7.182243e-04
PctLargHouseFam	0.028442315	0.00000000	0.000000000	0.0094807716	5.393102e-04
FemalePctDiv	0.028153141	0.00000000	0.000000000	0.0093843803	5.283996e-04
PctVacantBoarded	0.026867167	0.00000000	0.000000000	0.0089557224	4.812298e-04
NumIlleg	0.023814080	0.00000000	0.000000000	0.0079380266	3.780736e-04
HousVacant	0.022748240	0.00000000	0.000000000	0.0075827466	3.449883e-04
PctPolicMinor	0.020139887	0.00000000	0.000000000	0.0067132956	2.704100e-04
MalePctDivorce	0.019485816	0.00000000	0.000000000	0.0064952721	2.531314e-04
LemasTotalReq	0.012862603	0.00000000	0.000000000	0.0042875342	1.102977e-04
NumInShelters	0.007702610	0.00000000	0.000000000	0.0025675367	3.955347e-05
NumUnderPov	0.002403681	0.00000000	0.000000000	0.0008012269	3.851787e-06

(b) Jeu de données 2

	coeffs elastic net	coeffs iterative selection	coeffs it. sel. + elastic net	mean	std
MalePctDivorce	1.477667e-01	0.6694188	0.103490845	0.3068920354	1.981184e-01
whitePerCap	0.000000e+00	0.6644240	0.000000000	0.2214741323	2.943047e-01
racepctblack	1.825113e-01	0.3670430	0.113126845	0.2208937436	3.444654e-02
perCapInc	0.000000e+00	0.6173961	0.000000000	0.2057987057	2.541186e-01
PctKids2Par	-1.440412e-01	-0.1947389	-0.152515538	-0.1970985650	5.891689e-03
PctIlleg	1.842111e-01	0.2109531	0.193664436	0.1962762316	3.677998e-04
TotalPctDiv	0.000000e+00	-0.5630215	0.000000000	-0.1876738425	2.113288e-01
PctPersDenseHous	8.066319e-02	0.2905260	0.122190801	0.1644688633	2.470793e-02
NumStreet	1.363256e-01	0.1522578	0.132016464	0.1401999431	2.273720e-04
NumInShelters	8.804968e-02	0.1589798	0.000000000	0.0823431539	1.268613e-02
PctLargHouseFam	0.000000e+00	-0.2354706	0.000000000	-0.0784090192	3.697258e-02
LemasTotReqPerPop	9.459749e-02	0.1388150	0.000000000	0.0778041847	1.005785e-04
PctPolicAsian	3.185059e-02	0.1375936	0.055866451	0.0738254672	5.676487e-03
PctWorkMom	-4.272501e-02	-0.1303451	-0.033587459	-0.0688858837	5.707615e-03
PctHousOccup	-7.920080e-02	-0.0833648	-0.041174259	-0.0678938436	1.079333e-03
PctRecImmig5	4.133415e-07	0.1917881	0.000000000	0.0639295245	2.452175e-02
PctNotSpeakEnglWell	0.000000e+00	-0.1850756	0.000000000	-0.0616918824	2.283533e-02
MedOwnCostPctIncNoMtg	-6.312891e-02	-0.1012810	0.000000000	-0.0548030025	5.232963e-03
PolicReqPerOffic	2.542591e-02	0.1056191	0.027466378	0.0528371314	4.180987e-03
householdsize	0.000000e+00	0.1496788	0.000000000	0.0498929642	1.493585e-02
PctPolicBlack	3.126654e-02	0.1015510	0.010973379	0.0479303365	4.518685e-03
HousVacant	6.731928e-02	0.0695770	0.000000000	0.0456320981	3.125981e-03
pctWInvInc	-5.225122e-02	-0.0767168	0.000000000	-0.0429893434	3.071408e-03
racePctWhite	0.000000e+00	0.1519843	-0.039951720	0.0373442106	2.051161e-02
PctPolicHisp	4.788503e-03	0.1031095	0.000000000	0.0359660225	6.773848e-03
racePctAsian	0.000000e+00	-0.0974398	0.000000000	-0.0324799354	6.329677e-03
PctForeignBorn	3.281216e-02	0.0592590	0.002856580	0.0316425130	1.592680e-03
PctImmigRecent	-2.090537e-02	-0.0603750	0.000000000	-0.0270934725	1.880012e-03
PolicPerPop	-2.228281e-02	-0.0381559	-0.016366749	-0.0256018429	2.339092e-04
MedYrHousBuilt	0.000000e+00	-0.0595273	0.000000000	-0.0198424388	2.362334e-03
PctSameState85	0.000000e+00	0.0591844	0.000000000	0.0197281369	2.335196e-03
PctWOFullPlumb	0.000000e+00	-0.0545646	0.000000000	-0.0181854864	1.984272e-03
pctWFarmSelf	0.000000e+00	0.0540553	0.000000000	0.0180183752	1.947971e-03
MedRentPctHousInc	4.487399e-02	0.0539141	0.000000000	0.0179781388	2.335196e-03
PctWOFullPlumb	0.000000e+00	-0.0545646	0.000000000	-0.0181854864	1.984272e-03
pctWFarmSelf	0.000000e+00	0.0540553	0.000000000	0.0180183752	1.947971e-03
MedRentPctHousInc	4.487399e-02	0.0539141	0.000000000	0.0179781388	2.335196e-03
PctRecImmig10	2.756168e-02	0.0000000	0.000000000	0.0091872255	5.064307e-04
LemasSwornPT	1.150817e-02	0.0000000	0.015340701	0.0089496237	1.274878e-04
pctWPubAsst	2.540134e-02	0.0000000	0.000054879	0.0086494063	4.211899e-04
PctSameCity85	2.344541e-02	0.0000000	0.000000000	0.0078151365	3.664581e-04
PctPolicMinor	1.679673e-02	0.0000000	0.000000000	0.0055989090	1.880867e-04
pctWWage	-1.640975e-02	0.0000000	0.000000000	-0.0054699167	1.791199e-04
PctWorkMomYoungKids	-1.479485e-02	0.0000000	0.000000000	-0.0049316156	1.459250e-04
PolicCars	1.049364e-02	0.0000000	0.000000000	0.0034978784	7.341092e-05
pctUrban	9.750992e-03	0.0000000	0.000000000	0.0032503307	6.338790e-05
IndianPerCap	-9.004237e-03	0.0000000	0.000000000	-0.0030014124	5.405086e-05
LemasPctPolicOnPatr	5.520899e-03	0.0000000	0.000000000	0.0018402997	2.032022e-05
NumKindsDrugsSeiz	1.774750e-03	0.0000000	0.000000000	0.0005915835	2.099826e-06

(c) Jeu de données 3

Figure 12: Liste ordonnée des valeurs, moyennes et écarts-types du coefficient associé à chaque variable dans les modèles précédents (les coefficients absents ont une valeur de 0 dans tous les modèles)

(médiane des frais de propriétaire en pourcentage du revenu d'un foyer, négativement), "whitePerCap" (revenu des personnes blanches, négativement), "PctIlleg" (pourcentage d'enfants né de parents n'ayant jamais été mariés, positivement). Parmi les variables liées à la police uniquement présentes dans les jeux de données 2 et 3, on trouve dans les deux modèles "PctPolicAsian" (part asiatique des forces de police, positivement), PolicPerPop (nombre de policiers pour 100,000 personnes, négativement). Cependant comme énoncé précédemment on a des corrélations très fortes dans ce jeu de données, par exemple entre toutes les variables représentant le capital économique des populations, celles représentant la proportion d'immigrés et la racisation des habitants.

Les variables générales qui semblent avoir le plus d'influence sur le nombre de crimes violents par 100,000 habitants semblent être liées, sans ordre d'importance :

- la stabilité des liens familiaux, en particulier les variables "MalePctDivorce" (pourcentage d'hommes divorcés), "PctKids2Par" (pourcentage d'enfants vivant avec ses deux parents), "PctFam2Par" (pourcentage de familles avec enfants présentant deux parents), etc
- la situation économique, en particulier les variables "NumStreet" (nombre de sans-abris), "perCapInc" (revenu par habitant), "medIncome" (revenu moyen), "MedRent" (loyer moyen, qui est très corréllé positivement aux variables de revenus), etc
- absence de noirs et d'immigrés, en particulier les variables "racepctblack" (pourcentage de noirs), "PctForeignBorn" (pourcentage de personnes nées à l'étranger), "PctRecImmig5/8/10" (pourcentage d'immigrés

	Estimate	Pr(>|t|)
racepctblack	0.31281969	7.707916e-07
MalePctDivorce	0.27480483	2.196469e-06
NumStreet	0.24138179	6.473620e-05
PctWorkMom	-0.15590875	1.474232e-04
PctForeignBorn	0.23298201	1.620926e-04
PctEmploy	0.30701275	2.542964e-04
pctWWage	-0.31052828	2.895374e-04
MedOwnCostPctIncNoMtg	-0.11074995	3.186656e-04
PctHousOccup	-0.11607010	4.209519e-04
whitePerCap	-0.73643942	7.282029e-04
RentLowQ	-0.28040199	9.142691e-04
MalePctNevMarr	0.18330726	1.331937e-03
PctPersDenseHous	0.25284568	4.402140e-03
PctIlleg	0.17706825	5.296439e-03
perCapInc	0.64767388	5.985130e-03
HousVacant	0.11791783	1.354201e-02
PctLargHouseFam	-0.17895486	1.957155e-02
MedRent	0.21441193	2.281982e-02
PctHousOwnOcc	0.12837453	3.994922e-02

(a) Jeu de données 1

	Estimate	Pr(>|t|)
pctWInvInc	-2.48918782	4.457991e-09
PctNotHSGrad	-1.81112586	1.217309e-08
NumStreet	0.40292570	2.129761e-07
PctBornSameState	-1.13229332	2.203827e-07
MedOwnCostPctInc	-0.73189027	4.520012e-07
MedRent	4.17851026	6.412365e-06
pctWPubAsst	0.99256379	1.614498e-05
PctOccupMgmtProf	1.41138095	2.686698e-05
LemasSwFTFieldPerPop	2.54451458	3.525119e-05
PctForeignBorn	1.57777600	5.227244e-05
PctImmigRec5	-2.55259353	8.178226e-05
PolicPerPop	-2.13030943	1.194449e-04
PctRecImmig10	-2.54453763	1.365557e-04
PctSameState85	0.70297061	1.640443e-04
racePctAsian	-0.42864970	1.679086e-04
PolicCars	-0.47063098	2.489026e-04
indianPerCap	0.42168616	1.627711e-03
blackPerCap	-0.71439300	1.986876e-03
AsianPerCap	-0.42157170	3.254307e-03
MedNumBR	-0.18847571	3.558794e-03
MedYrHousBuilt	-0.33892953	4.314384e-03
PctImmigRec10	1.39855213	6.730074e-03
PctEmploy	1.03740053	6.910475e-03
PctFam2Par	1.75461044	7.042760e-03
PctIlleg	0.55326000	7.784692e-03
RentLowQ	-1.16921722	8.783941e-03
LemasPctPolicOnPatr	-0.29313752	1.013110e-02
whitePerCap	-0.66687450	1.186371e-02
medIncome	-1.43668182	2.464411e-02
(Intercept)	1.33256437	2.497551e-02
PctRecImmig5	1.08001715	2.616070e-02
pctWWage	-0.71244608	3.482095e-02
PersPerRentOccHous	-1.08775476	3.678513e-02
PctPolicAsian	0.15968419	4.508829e-02
PctHousOwnOcc	1.93489830	4.664628e-02

(b) Jeu de données 2

	Estimate	Pr(>|t|)
racepctblack	0.36704306	6.095796e-07
MalePctDivorce	0.66941858	1.740247e-05
PctWorkMom	-0.13034518	3.151508e-04
PctIlleg	0.21095313	9.297773e-04
MedOwnCostPctIncNoMtg	-0.10128210	1.042250e-03
TotalPctDiv	-0.56302153	1.128115e-03
PctLargHouseFam	-0.23549706	1.227578e-03
PctPersDenseHous	0.29055260	1.557314e-03
whitePerCap	-0.66442240	2.726547e-03
PctPolicAsian	0.13375936	4.462702e-03
PolicPerPop	-0.03815597	6.830964e-03
PolicReqPerOffic	0.10561911	7.443131e-03
perCapInc	0.61739612	8.094139e-03
PctPolicHisp	0.10310956	1.323897e-02
PctPolicBlack	0.10155110	1.711592e-02
PctNotSpeakEnglWell	-0.18507565	1.893450e-02
PctRecImmig5	0.19178816	2.309826e-02
PctHousOccup	-0.08330648	2.464276e-02
householdsize	0.14967889	2.527427e-02
NumStreet	0.15225780	3.565732e-02
PctKids2Par	-0.19473892	4.149383e-02
MedYrHousBuilt	-0.05952732	4.653226e-02

(c) Jeu de données 3

Figure 13: Liste des coefficients statistiquement significatifs (à 95%) et leur p-value associée calculés par les modèles linéaires aux variables sélectionnées itérativement

étant arrivé il y a moins de 5, 8 et 10 ans), etc

Chacune de ces variables générales influe négativement sur le taux de crimes violents. Ces variables générales peuvent également s'influencer entre elles (il y a par exemple des corrélations entre la situation économique et la race des habitants). Enfin il convient de rappeler qu'il faudrait des analyse sociologiques théoriques comme empiriques plus poussées pour tenter d'établir des liens de causalité.