Régression

Exercice 1 Prédiction de dépenses médicales
Le but de cet exercice est d'utiliser les données de patient pour prévoir les dépenses de soins médicaux.

1. Collecte des données
   Nous utiliserons des données simulées contenant des dépenses médicales hypothétiques pour des patients des États-Unis issues du livre [Lantz19] [1].
   Récupérez le fichier `insurance.csv` et enregistrez-le dans votre répertoire de travail.

2. Exploration et préparation des données
   — utilisation de la fonction `read.csv()` pour charger les données
   ```
   > assurance <- read.csv("insurance.csv")
   ```

   — utilisation de la fonction `str()` pour afficher le format des données
   ```
   > str(assurance)
   ```
   On en déduit que le fichier est composé de 1338 exemples de bénéficiaires. Les caractéristiques sont les suivantes :
   — `age` : entier indiquant l'âge du bénéficiaire
   — `sex` : genre : homme ou femme
   — `bmi` : indice de masse corporelle (poids/taille^2)
   — `children` : entier indiquant le nb d'enfants couverts par l'assurance
   — `smoker` : fumeur régulier ou non
   — `region` : lieu de résidence du bénéficiaire divisé en 4 zones géographiques : NE, SE, SO, NO
   La variable de notre modèle est `expenses` qui mesure les coûts médicaux annuels de chaque personne.

   — avant de construire le modèle de régression, il est souvent utile de vérifier la normalité. Même si une régression linéaire ne requière pas forcément une loi normale, l'ajustement du modèle sera meilleur lorsque c'est le cas. La fonction `summary` permet d'avoir les statistiques (min, max, moyenne, médiane, quartiles)
   ```
   > summary(assurance$expenses)
   ```

   — affichage de l'histogramme
   ```
   > hist(assurance$expenses)
   ```

   La distribution n'est pas idéale pour une régression linéaire. Le fait de connaître cette faiblesse nous permettra de concevoir un meilleur modèle après. Autre problème : les

1. Machine learning with R, Expert techniques for predictive modeling, B. Lantz, 3rd Edition, Packt Publishing, 2019

modèles de régression requièrent des variables numériques. Les variables `smoker` et `sex` sont divisées chacune en 2 catégories. La variable `region` a 4 niveaux. Pour avoir plus d'informations sur la distribution, on peut utiliser la fonction `table()` liste les catégories de la variable et compte le nombre de valeurs de chaque catégorie.

```
> table(assurance$region)
```

Que remarquez-vous ?

— matrice de corrélation pour étudier la dépendance entre les variables numériques en utilisant la fonction `cor`

```
> cor(assurance[c("age","bmi","children","expenses")])
```

Analysez les résultats obtenus.

3. entraînement du modèle

Pour ajuster un modèle de régression linéaire, on peut utiliser la fonction `lm()` dont la syntaxe est la suivante :

`m <- lm(dv ~ iv, data = mydata)` avec :

— `dv` est la variable à expliquer (endogène) que l'on souhaite modéliser dans le dataframe `mydata`

— `iv` est une formule de R spécifiant les variables explicatives (exogènes) du dataframe `mydata` à utiliser dans le modèle

— `data` spécifie le dataframe dans lequel les variables `dv` et `iv` se trouvent

La fonction retourne un modèle de régression `m` qui peut être utilisé pour faire des prédictions.

Les interactions entre les variables exogènes peuvent être spécifiées avec l'opérateur *.

Pour effectuer des prédictions, on utilisera la fonction `predict()` : `p <- predict(m,test)` avec :

— `m` : le modèle entraîné par la fonction `lm()`

— `test` est le dataframe contenant les données textes ayant les mêmes caractéristiques que les données d'entraînement utilisées pour la construction du modèle

La fonction retourne un vecteur contenant les valeurs prédites.

```
> assurance_model <- lm(expenses ~ age + children + bmi + sex + smoker + region, data = assurance)
```

Le caractère (.) peut être utilisé

```
> assurance_model <- lm(expenses ~ ., data = assurance)
```

Il suffit ensuite de taper le nom du modèle pour avoir les coefficients estimés :

```
> assurance_model
```

Interprétez les résultats obtenus.

4. évaluation des performances du modèle

Pour évaluer les performances du modèle, on peut utiliser la commande `summary()` :

```
> summary(assurance_model)
```

Interprétez les résultats obtenus, en particulier les résidus, la p-value et le coefficient de détermination (r^2).

5. amélioration des performances du modèle

Une différence entre le modèle de régression et les autres approches de machine learning est que la régression laisse à l'utilisateur la sélection des caractéristiques et de la spécification du modèle.

(a) spécification du modèle : ajout de relations non-linéaires
Dans une régression linéaire, la relation entre une variable endogène et les variables exogènes est supposée linéaire, ce qui n'est pas toujours le cas. Par exemple, l'effet de l'âge sur les dépenses médicales ne sera pas constant pour tous les âges : le traitement sera beaucoup plus cher pour des populations plus âgées.
Une équation de régression typique suit une forme de ce type : $y = \alpha + \beta_1 x$.
Pour prendre en compte la non-linéarité, on peut ajouter un terme d'ordre supérieur à l'équation de régression pour avoir un modèle polynomial : $y = \alpha + \beta_1 x + \beta_1 x^2$
Pour ajouter un âge non-linéaire au modèle, il suffit de créer une nouvelle variable :

```
> assurance$age2 <- assurance$age^2
```

(b) transformation : conversion des variables numériques en indicateur binaire
Dans certains cas, on peut penser que l'effet d'une caractéristique n'est pas cumulatif, mais qu'il a un impact uniquement à partir d'un certain seuil. Par ex, l'indice de masse corporelle (BMI) n'aura aucun impact sur les dépenses médicales d'individu de poids normal mais sera lié à des coûts plus élevés pour des personnes obèses (BMI>=30).
On peut modéliser cette relation en créant une variable binaire valant 1 pour BMI >=30 et 0 sinon. LA valeur beta estimée pour cette caractéristique binaire indiquera l'impact moyen sur les dépenses médicales pour des individus ayant un BMI de 30 ou plus, relativement à ceux avec un BMI inférieur à 30.
Pour créer cette caractéristique, on peut utiliser la fonction `ifelse()` :

```
> assurance$bmi30 <- ifelse(assurance$bmi >=30, 1, 0)
```

On peut ensuite inclure la variable `bmi30` dans le modèle soit en remplaçant la variable originale `bmi`, soit en la rajoutant.

(c) spécification du modèle : ajout des effets d'interaction
Pour l'instant, nous avons uniquement considéré la contribution individuelle de chaque caractéristique. Cependant certaines caractéristiques peuvent avoir un impact combiné sur la variable endogène. Par exemple, le fait de fumer et l'obésité peuvent avoir des effets nocifs séparément mais on peut penser que leur effet combiné peut avoir un effet pire que la somme de chaque élément séparé.
Quand deux variables ont un effet combiné, cela s'appelle une interaction.
Les effets de l'interaction sont spécifiées avec R en utilisant la syntaxe suivante : `expenses ~ bmi30*smoker`.
L'opérateur * est un raccourci R pour le modèle : `expenses ~ bmi30 + smokeryes + bmi30:smokeryes`

(d) modèle de régression amélioré
On va maintenant créer un modèle plus précis en ajoutant les spécifications suivantes :
— ajout d'un terme non-linéaire pour l'âge
— création d'un indicateur pour l'obésité
— spécification d'une interaction entre obésité et le fait de fumer
Entrainez le nouveau modèle :

```
> assurance_model2 <- lm(expenses ~ age + age2 + children + bmi + sex + bmi30*smoker + region, data = assurance)
```

Affichez et interprétez les résultats :

```
> summary(assurance_model2)
```

6. prédictions à partir du modèle

   On peut utiliser le modèle pour prédire les dépenses de futurs adhérents.

   Appliquez d'abord le modèle sur les données d'entraînement originales en utilisant la fonction `predict()`.

   ```
   > assurance$pred <- predict(assurance_model2,assurance)
   ```

   Les prédictions sont enregistrées dans un nouveau vecteur nommé `pred` dans le dataframe `assurance`. Vous pouvez ensuite calculer la corrélation entre les coûts réels et les coûts prédits :

   ```
   > cor(assurance$pred, assurance$expenses)
   ```

   Interprétez le résultat.

   On peut également tracer la relation entre les coûts actuels et les coûts prédits avec la commande suivante :

   ```
   > plot(assurance$pred, assurance$expenses)
   > abline(a=0, b=1, col="red", lwd=3, lty=2)
   ```

   La deuxième commande permet d'ajouter la ligne identité avec une pente de 1 et l'ordonnée à l'origine 0. Les paramètres `col`, `lwd` et `lty` permettent de changer respectivement la couleur de la ligne, la taille et le type.

   Interprétez les résultats obtenus.

   On va maintenant prédire les dépenses potentielles de nouveaux inscrits. Il suffit de fournir à la fonction `predict()` un dataframe avec la données du patient. Pour de nombreux patients, on peut créer un fichier CSV à charger dans R. Pour quelques patients, il suffit de créer un dataframe. Par exemple, pour un homme non fumeur de 30 ans en surpoids avec 2 enfants et de la région Northeast :

   ```
   > predict(assurance_model2,data.frame(age=30, age2=30^2, children=2, bmi=30,
   sex = "male", bmi30=1, smoker = "no", region = "northeast"))
   ```

   Testez ensuite pour une femme avec les mêmes caractéristiques, puis sans enfant. Comparez les résultats obtenus.