TD 1 REGRESSION

L'objectif de ce TD est d'étudier la prédiction des dépenses médicales de patients fictifs, en utilisant le langage R, sous Rstudio.

Collecte des données

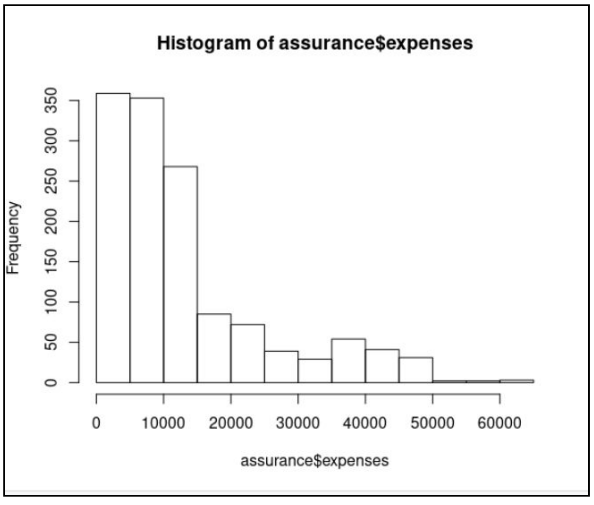
La collecte des données se fait via le fichier ci-joint, *insurance.csv.* Ce fichier contient les dépenses médicales hypothétiques de patients Américain, comme indiqué dans le sujet.
Pour pouvoir travailler sur ce fichier, nous devons l'ajouter à notre **répertoire de travail**, sur Rstudio. Pour se faire, il suffit de lire le fichier CSV. grâce à la fonction **read.csv**.

Exploration et préparation des données

Une fois nos données chargées, on peut utiliser la fonction **str(var_csv)** pour obtenir la forme des données de notre fichier.
Cette fonction retournera les différentes **caractéristiques** présentes dans le fichier, et le nombre d'**observations**. Pour les caractéristiques, on remarquera les différentes spécifications liée à chacune d'entre elles.
Cette fonction donne également accès à la **variable de notre modèle**. C'est la variable de classe, qui peut représenter le **vecteur cible** d'une classification.

Normalité

Ici, on cherche à établir une relation linéaire entre une variable et une autre, on peut parler d'**espérance conditionnelle** d'une variable par rapport à une autre. Comme indiqué, on peut avant tout essayé d'étudier la normalité de notre régression. Cela nous permet de constater la présence, ou non, de **valeurs aberrantes**, et la **distribution** des données, ce qui nous revient à déterminer la **loi normale** de notre modèle.
Pour la vérifier, on utilisera donc la fonction **summary(var_csv$varmodele)** qui nous permet d'obtenir une vue résumée d'une variable : ici la variable de notre modèle, suivi de la visualisation graphique de nos données à l'aide d'un histogramme : **hist(var_csv$varmodele)**.



On s'aperçoit dans notre cas que la distribution n'est pas normalisé, ce qui n'est pas un bon point pour notre modèle, car il ne semble pas homogène. Il faudra donc améliorer notre modèle.

Cependant, cette représentation ne suffit pas, puisqu'elle ne prend en compte que les valeurs numériques de notre modèle. Certaines de nos variables ne possédant pas de valeurs numériques, il nous faut pouvoir les observer.

Pour se faire, nous allons utiliser la fonction **table(var_csv$variable_a_etudier)**. Ici, on veut obtenir des informations sur les variables *smoker*, *sex*, et *region* par exemple.

```
> table(assurance$region)

northeast northwest southeast southwest
      324       325       364       325
> table(assurance$smoker)

  no  yes
1064  274
> table(assurance$sex)

female   male
   662    676
```

On peut donc voir ci-contre que la population semble assez bien répartie entre chaque région pour la première caractéristique, et au niveau du sexe. Le nombre de non-fumeur lui, est beaucoup plus important que celui de fumeurs : on a donc une **mauvaise** distribution des données à ce niveau.

Par la suite, nous allons étudier la **dépendance** entre les variables numérique, grâce à une **matrice de corrélation**. Cela nous permet d'obtenir les **coefficients de corrélation** entre chaque variable, c'est-à-dire leur liaison l'une par rapport à l'autre.
On peut donc étudier la matrice de corrélation entre les variables numériques de notre système, grâce à la fonction **cor(var_csv[c(var1,var2,...)])**. Ici on travaillera sur les variables *age, bmi, childre,* et *expenses*.

Note : La visualisation d'une matrice de corrélation sera possible, si la matrice est composée de valeurs numériques uniquement, grâce à la fonction ***corrplot***.

```
> cor(assurance[c("age", "bmi", "children", "expenses")])
               age        bmi   children   expenses
age      1.0000000 0.10934101 0.04246900 0.29900819
bmi      0.1093410 1.00000000 0.01264471 0.19857626
children 0.0424690 0.01264471 1.00000000 0.06799823
expenses 0.2990082 0.19857626 0.06799823 1.00000000
```

Pour interpréter les valeurs obtenues, on notera que plus un coefficient est proche de **0**, plus les valeurs seront **linéairement indépendantes.** On pourra parler de **liaison linéaire** si le coefficient s'approche de **1.** On peut, d'ailleurs, trouver des corrélation **négative**, cela signifiera simplement que notre deuxième variera en sens inverse de la première, mais qu'elles sont bien liées. La diagonale sera forcément constituée de **1** puisque la corrélation d'une variable avec cette même variable sera parfaite.

Pour cet exemple, la plus grande dépendance se trouve entre la **dépense et l'âge**, à environ 0.29, ce qui semble cohérent. La plus petite se trouve entre l'indice de masse corporelle, et le nombre d'enfants d'une personne.

Entraînement du modèle

Maintenant que notre modèle à été analysé, et que nous avons pu observer ses problèmes de normalités, nous pouvons essayer de l'améliorer.
Pour l'ajuster, nous utiliserons la fonction **lm**. Cette fonction nous permet de calculer les modèles correspondant à nos données. Cette fonction est définie par différents paramètres , et est représenté sous la forme : **m <- lm(dv ~ iv, data = mydata)**

- **dv** qui correspond à la variable que l'on cherche à expliquer, celle que l'on souhaite traiter, qui se trouve dans le data frame utilisé. Ce sera généralement la variable du modèle.
  Ici, dv représente la variable 'expenses' : on cherche à étudier les dépenses médicales.
- **m** représente le modèle de régression que l'on va créer pour stocker le résultat de la fonction **lm** pour notre modèle. Ce modèle pourra être utilisé par la suite pour faire des prédictions.
- **iv** permet de définir les variables explicatives de la variable du modèle. On pourra donc, soit ajouter les variables à étudier les unes à la suite des autres, soit si l'on veut travailler sur toutes nos variables, utiliser le caractère "**.**" .
  S'il existe des interactions entre ces variables dites "exogènes", on pourra utiliser l'opérateur *****.
- **data** spécifie le data frame **mydata** sur lequel on souhaite travailler, et où se trouve les données **dv** et **iv**.

Pour effectuer la prédiction de notre modèle, l'utilisation de la fonction **predict** sera nécessaire. Cette fonction s'utilise sous la forme : **p <- predict(m, test)**.

- Ici, **m** représentera le modèle que l'on aura, au préalable, entraîné avec la fonction lm,
- **test** sera le data frame possédant les mêmes caractéristiques que les données d'entraînement utilisé pour construire le modèle.

Cette fonction retourne un vecteur constituée des valeurs prédites.

Maintenant que nous connaissons précisément le fonctionnement de ces fonctions, nous pouvons les utiliser dans le cadre de notre exercice. Pour se faire, nous allons étudier le résultat de chacunes d'entre elles :

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

Call:
lm(formula = expenses ~ ., data = assurance)

Coefficients:
    (Intercept)              age          sexmale              bmi         children        smokeryes  regionnorthwest  regionsoutheast
      -11941.6            256.8            -131.4            339.3            475.7          23847.5           -352.8           -1035.6
regionsouthwest
        -959.3
```

Nous avons donc commencé par définir le **modèle de régression** de notre système. L'appel de la fonction de cette manière ne nous fournit pas toutes les informations nécessaires, mais nous pouvons commencer de les interpréter.

En premier, nous pouvons apercevoir le rappel du modèle qui a été appelé avec le "**Call**". Ici, on travaille bien sur la variable à expliquer **expenses**, sur le data frame **assurance**.
La partie **coefficients** représente les valeurs que le modèle estime, ce qui peut nous donner de premières brèves informations concernant les estimations de chacun des paramètres. La première donnée : **intercept**, correspond à l'ordonnée à l'origine.
La deuxième sur la colonne "**age**", nous permet d'obtenir le **coefficient de la pente**. On peut donc dire pour notre cas que lorsque l'âge augmente d'une unité, les dépenses

augmentent de 256.8 unité. Pour interpréter plus précisément ces coefficients, nous pouvons utiliser la fonction **summary** afin d'évaluer les performances de notre modèle.

Evaluation des performances du modèle

```
> summary(assurance_model)

Call:
lm(formula = expenses ~ ., data = assurance)

Residuals:
     Min       1Q   Median       3Q      Max
-11302.7  -2850.9   -979.6   1383.9  29981.7

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -11941.6      987.8 -12.089  < 2e-16 ***
age                256.8       11.9  21.586  < 2e-16 ***
sexmale           -131.3      332.9  -0.395 0.693255
bmi                339.3       28.6  11.864  < 2e-16 ***
children           475.7      137.8   3.452 0.000574 ***
smokeryes        23847.5      413.1  57.723  < 2e-16 ***
regionnorthwest   -352.8      476.3  -0.741 0.458976
regionsoutheast  -1035.6      478.7  -2.163 0.030685 *
regionsouthwest   -959.3      477.9  -2.007 0.044921 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6062 on 1329 degrees of freedom
Multiple R-squared:  0.7509,    Adjusted R-squared:  0.7494
F-statistic: 500.9 on 8 and 1329 DF,  p-value: < 2.2e-16
```

On peut donc voir ci-dessus que l'exécution de la fonction summary nous a permis de compléter les informations précédentes. On peut apercevoir tout d'abord qu'une partie s'est ajoutée, "**Residuals**".
Cette partie permet d'évaluer rapidement la normalité des **résidus** (approximation des erreurs connues). Si ces résidus sont distribués autour d'une loi normale, la **médiane** doit être autour de **0**, et les valeurs de **Q1** et **Q3** doivent être proches. Dans notre cas, on peut donc conclure que nos résidus ne sont pas définis par une loi normale : ce qui confirme notre hypothèse de base.
On a ensuite plus d'informations sur la partie "**Coefficients**". Notre première colonne correspond à l'**estimation des coefficients des paramètres**, la deuxième à l'**estimation de** leur **erreur standard**, la troisième est **statistique T** (rapport entre la valeur absolue de la pente, et l'erreur standard), et la dernière est la **p-value** du test. Cette valeur évalue l'égalité à 0 des coefficients, si elle est inférieure à 0.05 on pourra spécifier que le **paramètre** est donc significativement **différent de 0** (c'est le cas ici).
La donnée la plus importante reste donc le **coefficient de la pente** que l'on a étudié précédemment. On peut s'y intéresser plus en détail, car le fait de constater que ce

coefficient est fort, ne signifie pas qu'il est différent de 0. La significativité de la pente dépend de la **dispersion des points autour de la droite de régression**. C'est là que la **p-value** intervient : si elle est inférieure à 0.05, on peut conclure d'un lien entre la variable obtenue et la variable prédictive. Le sens de cette relation sera donné par le **signe du coefficient** (*s'il est positif, la relation linéaire est croissante, s'il est négatif, alors la relation linéaire est décroissante*). On pourra donc affirmer que la relation entre ces variables est **significative**.

La partie "**Residual standard error**" retourne également certaines informations. Elle représente l'écart-type des résidus, ainsi que leur degré de liberté.
Les parties "**Multiple R-squared**" et "**Adjusted R-squared**" sont les coefficients de corrélation. Plus ils sont proches de 1, meilleur est notre modèle. Ici, notre modèle se rapproche de 1, on peut le considérer comme bon.
La dernière ligne, "**F-statistics**" possède une p-value que l'on peut étudier de la même manière.

Pour conclure et d'après l'étude des différents paramètres, on peut spécifier que, bien que notre modèle ne soit pas distribué autour d'une loi normale, il semble être bon bien qu'imparfait. Il nécessite donc quelques améliorations.

Amélioration des performances du modèle

Puisque l'on se trouve dans un modèle de régression, nous allons donc devoir sélectionner les caractéristiques et la spécification du modèle.
La première étape sera la **spécification**, qui correspond à l'**ajout de relations non-linéaire**.
La relation entre les variables à expliquer et les variables expliquées doit être **linéaire**, comme on l'a vu précédemment. L'exemple qui nous est donné est celui de l'effet de l'âge sur les dépenses médicales : le traitement sera beaucoup plus cher pour des populations plus âgées.
On souhaite désormais prendre en compte la **non-linéarité** dans notre régression. Pour se faire, dans un modèle polynomial, il suffit d'ajouter un terme d'ordre supérieur à notre équation. Dans notre cas, on ajoute donc un âge non-linéaire, comme spécifié, en créant une variable avec la syntaxe : **assurance$age2 <- assurance$age^2**

La deuxième étape est désormais la **transformation** qui consiste en la **conversion des variables numériques en indicateur binaire**.
On nous précise que certaines caractéristiques n'auront un **impact qu'à partir d'un certain seuil**. L'exemple qui nous est donnée est celui de l'indice de masse corporelle, appelé BMI ici. L'impact sur les dépenses médicales n'a lieu que lorsque l'IMC dépasse les 30, ce qui correspond à des coûts plus élevés pour une personne obèse.
La modélisation de cette relation sous la forme que l'on recherche, consiste en la création d'une variable binaire, sous la forme d'un **indicateur**, qui sera égale à 1 si l'IMC est >= à 30, sinon à 0 : ce qui correspond à la condition qui nous intéresse concernant les dépenses.
On nous indique que la ligne à utiliser pour prendre créer cet indicateur est la suivante : **assurance$bmi30 <- ifelse(assurance$bmi >= 30, 1, 0)**.

La troisième étape est la **spécification du modèle**, c'est l'ajout des effets d'interactions, qui va au-delà de la considération individuelle de chaque caractéristique. Dans cette partie,

nous allons prendre en considération les paramètres qui, pris ensemble, peuvent avoir un effet combiné.
Dans le sujet, il nous est donné l'exemple de fumer et de l'obésité, qui peuvent avoir un effet encore plus nocif lorsqu'on les cumule. La notion d'effet combiné s'appelle : une **interaction**.
On peut donc spécifier que, pour notre modèle, il existe des interactions entre certains paramètre : **expenses ~ bmi30*smoker**.

Ces différentes étapes nous ont permis d'obtenir un modèle plus précis en ajoutant diverses spécifications : c'est l'**amélioration de notre modèle de régression**.
Maintenant que notre modèle est amélioré et donc plus complet, nous pouvons l'entraîner : **assurance_model2 <- lm(expenses ~ age + age2 + children + bmi + sex + bmi30*smoker + region, data = assurance)**.

Une fois le modèle entraîné, voici les résultats qui en ressortent :

```
Call:
lm(formula = expenses ~ age + age2 + children + bmi + sex + bmi30 *
    smoker + region, data = assurance)

Residuals:
     Min       1Q   Median       3Q      Max
-17297.1  -1656.0  -1262.7   -727.8  24161.6

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       139.0053  1363.1359   0.102 0.918792
age               -32.6181    59.8250  -0.545 0.585690
age2                3.7307     0.7463   4.999 6.54e-07 ***
children          678.6017   105.8855   6.409 2.03e-10 ***
bmi               119.7715    34.2796   3.494 0.000492 ***
sexmale          -496.7690   244.3713  -2.033 0.042267 *
bmi30            -997.9355   422.9607  -2.359 0.018449 *
smokeryes       13404.5952   439.9591  30.468  < 2e-16 ***
regionnorthwest  -279.1661   349.2826  -0.799 0.424285
regionsoutheast  -828.0345   351.6484  -2.355 0.018682 *
regionsouthwest -1222.1619   350.5314  -3.487 0.000505 ***
bmi30:smokeryes 19810.1534   604.6769  32.762  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4445 on 1326 degrees of freedom
Multiple R-squared:  0.8664,    Adjusted R-squared:  0.8653
F-statistic: 781.7 on 11 and 1326 DF,  p-value: < 2.2e-16
```

On peut donc voir que les résidus ne sont toujours pas distribués autour d'une loi normale. De la même façon, notre **p-value** est toujours identique.
Cependant le coefficient de la pente a **baissé** pour notre relation non-linéaire : **age2**. On peut désormais dire que, lorsque l'âge augmente d'une unité, les dépenses augmentent de 3.7307 unités. L'écart-type entre les erreurs a lui aussi réduit.

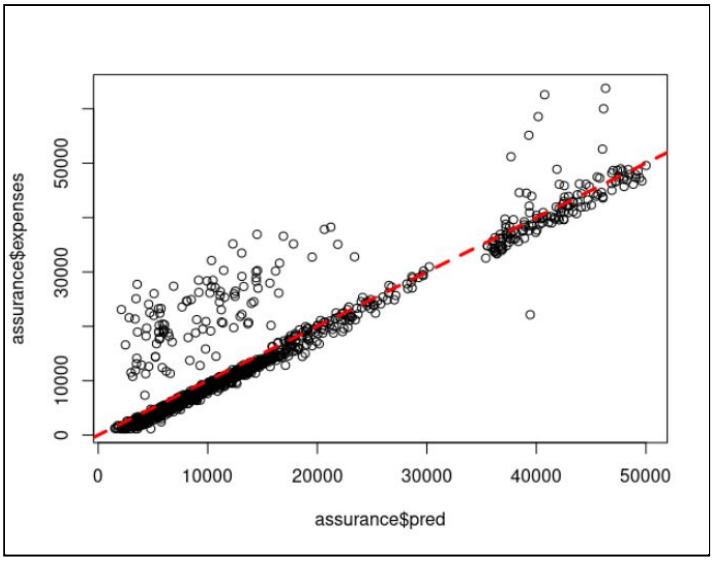
Les données correspondant à l'interaction que nous avons spécifiée se sont également ajoutée, nous permettant d'obtenir des **données concernant le lien entre obésité et fumeur**.

Une bonne nouvelle est que, les variable "**Multiple R-squared**" et "**Adjusted R-squared**" qui correspondent aux coefficients de corrélation se rapprochent de 1, ce qui indique bien que notre modèle **s'est amélioré** !

Prédictions à partir du modèle

Notre modèle étant donc plus fiable désormais, nous pouvons essayer de prédire les dépenses des futurs adhérents. Pour se faire, nous allons utiliser le modèle sur les données de l'entraînement **original :**
assurance$pred <- predict(assurance_model2, assurance) Cette ligne spécifie que les prédictions engendrées seront enregistrées dans le nouveau vecteur **pred** dans le data frame de base : **assurance**.

Une fois la prédiction prête, nous pouvons déterminer la corrélation entre les coûts réels et les coûts prédits : **cor(assurance$pred, assurance$expenses)**. Nous obtenons alors la valeur : 0.9307999. On peut donc ici parler de liaison linéaire, puisque le coefficient de corrélation se rapproche de 1. Nos coûts prédit se rapprochent donc assez bien des coûts réels.
Pour s'en assurer, nous pouvons visualiser ces données afin de mieux les interpréter. Cette visualisation s'obtient grâce à **plot(assurance$pred, assurance$expenses)**. Cependant, on précise que **abline(a=0, b=1, col="red", lwd=3, lty=2)**, ce qui nous permet d'ajouter la **ligne identité** qui est la ligne avec une pente de 1 en partant de l'origine 0. Côté visualisation pure, cette ligne nous permet également de choisir la taille, la couleur, et le type de ligne que l'on veut obtenir sur le graphique. Le résultat que l'on obtient avec ces commandes est le suivant :



Les données prédites semblent bien se retrouver **autour de la ligne identité**, de la même manière que les valeurs réelles. On notera cependant cer**taines valeurs** sortant de ce cadre, dispersé plus loin autour de la ligne identité, ce qui laisse apercevoir que notre modèle **peut toujours être amélioré**.

Puisque notre modèle semble obtenir des valeurs cohérentes, nous pouvons essayer de **prédire les dépenses potentielles de nouveaux inscrits**. Pour se faire, nous pouvons comme indiqué, soit choisir de charger un fichier CSV correspondant à plusieurs patients, ou

bien en générer quelques uns à la main pour en conclure quelque chose : c'est la solution que nous allons prioriser pour ce TD.
Le principe est donc, de créer les patients donné, et d'en interpréter leur résultat pour s'assurer qu'ils ne soient pas farfelus :
predict(assurance_model2, data.frame(age=30, age2=30^2, children=2, bmi=30, sex="male", bmi30=1, smoker="no", region="northeast")).

Nous avons donc testé sur un homme non fumeur de 30 ans en surpoids, avec 2 enfants, de la région Northeast, une femme avec les mêmes caractérsirtiques, et une autre identique mais sans enfant. Les résultats obtenus sont les suivants :

```
> # Prédiction de données de nouveaux inscrits
> #1 - Homme non fumeur de 30 ans en surpoids avec 2 enfants 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"))
       1
5973.774
> # 2 - Femme avec les mêmes caractéristiques que le précédent
> predict(assurance_model2, data.frame(age=30, age2=30^2, children=2, bmi=30, sex="female", bmi30=1, smoker="no", region="northeast"))
       1
6470.543
> # 3 - Femme avec les mes caractéristiques que la précédente mais sans enfants
> predict(assurance_model2, data.frame(age=30, age2=30^2, children=0, bmi=30, sex="female", bmi30=1, smoker="no", region="northeast"))
      1
5113.34
```

On peut donc voir que la femme possédant les mêmes caractéristiques que l'homme aura, d'après les prédictions de notre modèle, **plus de frais médicaux**. Nous pouvons imaginer que cela peut être dû à la naissance de ses 2 enfants par exemple, et ce que cela peut engendrer pour elle.
Pour la femme dans le même cas, mais sans enfant, on peut voir que ses frais sont **beaucoup moins importants** a contrario.

CONCLUSION
Ce TD nous a permis de nous initier à R, ainsi qu'à la régression linéaire. Nous avons pu, grâce à une base de donnée construite, créer et étudier un tel modèle afin de l'améliorer, dans l'objectif de pouvoir l'utiliser pour des prédictions futures. Cela nous a permis de voir qu'avec de simples spécifications, nous pouvons soigner un modèle, pour l'entraîner et le rendre meilleur.
Nous avons également pu travailler sur les prédictions de notre modèle, et étudier ces prédictions.

RÉFÉRENCES BIBLIOGRAPHIQUES

Statistiques et Logiciel R
Bioinfo-fr