



جامعة محمد الأول وجدة

UNIVERSITE MOHAMMED PREMIER OUJDA

+٢٠٦٨٠٥٤٣٨٩٣٣٩٨٠٥٤٣٠٥٨



المدرسة الوطنية للعلوم التطبيقية

ENSA +٢٠٦٨٠٥٤٣٨٩٣٣٩٨٠٥٤٣٠٥٨

École Nationale des Sciences Appliquées

# Rapport du projet modelisation statistique

**Sujet:** Prédiction des Coûts d'Assurance Médicale : Analyse et Modélisation à l'aide de la Régression Linéaire Multiple

Réalisé par:

FARAH Zineb

SAGAOUI Fatima

LAMRAOUI Hajar

Encadré par Mme: Rachida Mehdi

## **REMERCIEMENT:**

À notre chère professeur Rachida ELMEHDI,

Merci infiniment pour votre soutien et vos conseils tout au long de ce semestre. Votre disponibilité et votre approche pratique ont grandement facilité notre travail sur ce projet. Grâce à vos explications claires et à votre accompagnement, nous avons pu surmonter les difficultés et avancer avec confiance. Ce projet reflète tout ce que nous avons appris et réalisé grâce à votre précieuse guidance. Avec toute notre gratitude et notre respect, encore merci pour tout ce que vous avez fait pour nous.

# SOMMAIRE

---

1. <u>Introduction</u> .....	5
2. <u>CADRE THEORIQUE</u> .....	6
2.1 <u>Définition et Formule</u> .....	6
2.2 <u>Les hypothèses de base de la régression linéaire multiple</u> .....	6
2.3 <u>Matrice variance-covariance</u> .....	7
2.4. <u>Matrice De correlation</u> .....	8
2.5. <u>Coefficient de determination</u> .....	8
2.6. <u>Tests d'hypothese</u> .....	9
2.6.1 <u>Tests d'hypothèse sur un seul paramètre du modèle (test de student)</u> .....	10
2.6.2 <u>Test d'hypothèse global du modèle (test de Fisher)</u> .....	10
2.7. <u>Probleme de multi-colinéarité</u> .....	11
2.8. <u>Intervalle de confiance</u> .....	11
2.9. <u>Analyse des residus</u> .....	12
2.10. <u>Analyse de la Variance (ANOVA)</u> .....	12
3. <u>CADRE PRATIQUE</u> .....	14
3.1 <u>Overview sur le projet</u> .....	14
• <u>Rapport sur le Dataset</u> .....	14
• <u>Choix du language Python et les bibliothèques</u> .....	15
3.2. <u>le code</u> .....	17
1. <u>Importation des données</u> .....	17

# SOMMAIRE

---

2. <u>Analyse exploratoire des données</u> .....	18
3. <u>Nettoyage du dataset</u> .....	19
3. <u>Visualisation des donnees</u> .....	22
4. <u>Regression sur chaque variable</u> .....	30
5. <u>Preparation des donnees pour la regression</u> .....	39
6. <u>Regression sur toutes les variables</u> .....	40
6. <u>Regression sur toutes les variables(from scratch)</u> .....	47
8. <u>Comparaison des modèles</u> .....	50
9. <u>Problème de Colinéarité</u> .....	51
4. <u>CONCLUSION</u> .....	53

# **1. INTRODUCTION:**

Dans le domaine de l'assurance, la détermination précise des frais à facturer aux clients est essentielle pour garantir une rentabilité tout en restant compétitif. Les frais d'assurance sont influencés par une multitude de facteurs tels que l'âge, le sexe, l'indice de masse corporelle (IMC), le nombre d'enfants à charge, le statut de fumeur et la région géographique. Ce projet a pour objectif de modéliser ces relations complexes en utilisant la régression linéaire multiple, une méthode statistique puissante permettant de prédire une variable cible à partir de plusieurs variables explicatives.

Le dataset utilisé dans ce projet contient des informations sur 1338 assurés, offrant une base solide pour analyser les tendances et identifier les principaux facteurs contribuant aux frais d'assurance. À travers une approche méthodique incluant l'analyse exploratoire des données, la préparation des variables, la construction d'un modèle de régression linéaire multiple et son évaluation, ce projet vise à fournir des prédictions précises tout en offrant une interprétation claire des coefficients obtenus.

En plus de la modélisation, ce projet met en lumière les défis associés aux données réelles, tels que les corrélations entre variables ou la normalisation des échelles. Les résultats obtenus permettront non seulement de mieux comprendre les mécanismes sous-jacents des frais d'assurance, mais également de proposer des recommandations basées sur des faits, utiles pour les assureurs dans leurs prises de décision.

## **2.CADRE THEORIQUE:**

### **2.1 DÉFINITION ET FORMULE**

La régression linéaire multiple est une méthode statistique qui vise à modéliser la relation entre une variable dépendante (ou cible) et plusieurs variables indépendantes (ou explicatives). Contrairement à la régression linéaire simple, qui ne prend en compte qu'une seule variable explicative, la régression linéaire multiple permet d'examiner l'effet simultané de plusieurs facteurs sur la variable cible.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_j x_{ij} + \cdots + \beta_p x_{ip} + \varepsilon_i , \quad i = 1, 2, \dots, n$$

- $Y$  : Variable dépendante (valeur prédictive ou cible).
- $X_1, X_2, \dots, X_p$ : Variables indépendantes (prédicteurs).
- $\beta_0$  : Ordonnée à l'origine (interception).
- $\beta_1, \beta_2, \dots, \beta_p$ : Coefficients de régression, représentant l'effet de chaque prédicteur sur  $Y$ .
- $\varepsilon$ : Terme d'erreur (bruit aléatoire ou facteurs non pris en compte).

L'objectif principal est de minimiser l'erreur entre les valeurs observées et les valeurs prédictives en ajustant les coefficients  $\beta$ .

### **2.2 LES HYPOTHÈSES DE BASE DE LA RÉGRESSION LINÉAIRE MULTIPLE**

- **Hypothèses de Base de la Régression Linéaire Multiple**

Pour garantir la validité des résultats et des prédictions, la régression linéaire multiple repose sur plusieurs hypothèses fondamentales :

- **Relation linéaire**

Il doit exister une relation linéaire entre la variable dépendante  $Y$  et chaque variable explicative  $X_i$ . Cela signifie que le modèle doit capturer des tendances linéaires dans les données, sans quoi les résultats pourraient être biaisés.

- **Indépendance des erreurs (autocorrélation)**

Les termes d'erreur  $\epsilon$  doivent être indépendants les uns des autres. Cette hypothèse est particulièrement importante dans les données temporelles où une autocorrélation peut exister.

- **Homoscédasticité**

La variance des termes d'erreur doit être constante pour toutes les valeurs des variables explicatives. Si la variance n'est pas constante (hétérosécédasticité), cela peut affecter l'interprétation des coefficients.

- **Normalité des erreurs**

Les termes d'erreur  $\epsilon$  doivent suivre une distribution normale. Cette hypothèse est nécessaire pour effectuer des tests d'hypothèses fiables et construire des intervalles de confiance.

- **Absence de multicolinéarité**

Les variables explicatives  $X_1, X_2, \dots, X_p$  doivent être indépendantes les unes des autres. Si une forte corrélation existe entre deux ou plusieurs variables explicatives, cela peut compliquer l'estimation des coefficients  $\beta$  et leur interprétation.

## 2.3 MATRICE VARIANCE-COVARIANCE

Une matrice de variance-covariance est une matrice carrée qui résume la variabilité et la relation linéaire entre plusieurs variables aléatoires dans un ensemble de données. Elle est utilisée en statistiques pour analyser la dispersion des données et comprendre comment les variables sont corrélées.

Soit un vecteur aléatoire  $X = [X_1, X_2, \dots, X_n]^T$ , où  $X_1, X_2, \dots, X_n$  sont des variables aléatoires. La matrice de variance-covariance est définie comme :

$$V = \text{Var}(X) = E[(X - E[X])(X - E[X])^T]$$

Structure de la matrice

- Les éléments diagonaux ( $\sigma_{ii}$ ) représentent les variances des variables ( $\text{Var}(X_i)$ ).
- Les éléments hors diagonale ( $\sigma_{ij}$ ) représentent les covariances entre les variables  $X_i$  et  $X_j$  ( $\text{Cov}(X_i, X_j)$ ).

$$V = \begin{pmatrix} V(X_1) & Cov(X_1, X_2) & \dots & Cov(X_1, X_n) \\ Cov(X_2, X_1) & V(X_2) & \dots & Cov(X_2, X_n) \\ \vdots & & & \vdots \\ Cov(X_n, X_1) & Cov(X_n, X_2) & \dots & V(X_n) \end{pmatrix}.$$

## 2.4. MATRICE DE CORRELATION:

La matrice de corrélation mesure les relations linéaires entre plusieurs variables. Elle est normalisée, ce qui permet de comparer les relations entre variables indépendamment de leurs unités ou échelles. Voici ses propriétés et sa structure.

Pour p variables  $X_1, X_2, \dots, X_p$ , la matrice de corrélation R est une matrice carrée  $p \times p$ :

$$R = \begin{pmatrix} 1 & \cdots & r_{1j} & \cdots & r_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{j1} & \cdots & 1 & \cdots & r_{jp} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{p1} & \cdots & r_{pj} & \cdots & 1 \end{pmatrix}$$

$$R = S^{-1} \Sigma S^{-1} = S^{-1} \frac{X_c^t X_c}{n} S^{-1}$$

Les éléments diagonaux valent 1 : la corrélation d'une variable avec elle-même.

Les éléments hors diagonale  $r_{ij}$ : représentent les coefficients de corrélation entre les variables  $X_i$  et  $X_j$ .

## 2.5. COEFFICIENT DE DETERMINATION

Le coefficient de détermination est défini par le rapport entre la variance de régression et la variance totale tel que

$$R^2 = \frac{SCE}{SCT} = 1 - \frac{SCR}{SCT}$$

Sachant que :

- **SCE** : la somme des carrés des erreurs
- **SCR**: la somme des carrés de la régression
- **SCT= SCR+SCE** : la somme des carrés totale

Ce coefficient varie entre 0 et 1, où une valeur proche de 1 indique que le modèle explique une grande partie de la variabilité de la variable dépendante. Dans certains cas, il est plus judicieux d'utiliser le coefficient de détermination ajusté ( $R^2_{ajusté}$ ) plutôt que le coefficient de détermination classique ( $R^2$ ) en raison des différences fondamentales dans leurs propriétés. Pour comprendre quand opter pour l'un plutôt que pour l'autre, il est crucial d'examiner et de comparer leurs caractéristiques spécifiques.

$$R^2_{adj} = 1 - (1 - R^2) \frac{n - 1}{n - p - 1}$$

## 2.6. TESTS D'HYPOTHÈSE:

Les tests d'hypothèse permettent de prendre des décisions statistiques en vérifiant si les relations observées dans les données sont significatives ou dues au hasard. Pour la régression linéaire multiple, les tests de Student et tests de Fisher sont essentiels pour évaluer la contribution des variables explicatives et la qualité globale du modèle.

Un test d'hypothèse repose sur deux étapes principales :

- Hypothèse nulle ( $H_0$ ) : Représente l'absence d'effet ou de relation.
- Hypothèse alternative ( $H_1$ ) : Représente une relation ou un effet significatif.

L'objectif est de rejeter ou non  $H_0$  en fonction de la statistique de test et d'un seuil de signification ( $\alpha=0.05$  en général).

## 2.6.1 TESTS D'HYPOTHÈSE SUR UN SEUL PARAMÈTRE DU MODÈLE (TEST DE STUDENT)

**Hypothèses :**

- $H_0 : \beta_i = 0$  (la variable  $X_i$  n'a aucun effet significatif sur  $Y$ ).
- $H_1 : \beta_i \neq 0$  (la variable  $X_i$  a un effet significatif).

**Méthode :**

Calculer la statistique de test  $t_{cal}$ :

$$t_{cal} = \frac{\hat{\beta}_j - \beta_{j0}}{\hat{s}_{\hat{\beta}_j}} = \frac{(\hat{\beta}_j - \beta_j) / s_{\hat{\beta}_j}}{\sqrt{\chi^2_{(n-(p+1))} / (n - (p + 1))}}$$

Comparer la valeur  $t_{cal}$  avec une table  $t$  ou calculer la p-value.

**Décision :**

- Si  $p < \alpha$  (par exemple  $p < 0.05$ ), rejetez  $H_0$  : la variable  $X_i$  est significative.
- Sinon, ne rejetez pas  $H_0$ .

## 2.6.2 TEST D'HYPOTHÈSE GLOBAL DU MODÈLE (TEST DE FISHER)

- **Hypothèses :**

- $H_0$  : Tous les coefficients ( $\beta_1, \beta_2, \dots, \beta_p$ ) sont nuls, donc le modèle ne contribue pas à expliquer la variable cible.
- $H_1$  : Au moins un des coefficients est différent de 0.

**Méthode :**

Calculer la statistique de test  $F_{cal}$ :

$$F_{cal} = \frac{\frac{SCReg}{p}}{\frac{SCE}{n-(p+1)}}$$

Comparer la statistique  $F_{cal}$  avec la valeur critique de la table F ou utiliser la p-value.

**Décision :**

- Si  $p < \alpha$  (par exemple  $p < 0.05$ ), rejetez  $H_0$  : le modèle est globalement significatif.
- Sinon, ne rejetez pas  $H_0$ .

## 2.7. PROBLEME DE MULTI-COLINÉARITÉ

La matrice sans la première colonne est supposée de plein rang, c.à.d. de dimension . Donc, les variables explicatives sont supposées indépendantes et l'existence de forte dépendance entre elles fausse l'estimation des paramètres. Une première étape est de dresser la matrice de corrélation entre ces variables explicatives pour détecter les fortes corrélations afin de faire apparaître les interactions deux à deux possibles. Pour tester l'interaction entre trois variables explicatives, on régresse l'effet combiné de deux sur la troisième mais ceci devient lourd surtout lorsque le nombre de variables est important.

## 2.8. INTERVALLE DE CONFIANCE

Lorsqu'on souhaite estimer les valeurs des paramètres inconnus, une simple estimation ponctuelle ne suffit généralement pas pour prendre des décisions robustes. C'est pourquoi on préfère construire des intervalles de confiance. Pour ce faire, il est essentiel de définir la distribution des estimations des coefficients. En pratique, on suppose généralement un niveau de confiance de 95% (ce qui correspond à  $\alpha = 0,05$ ). Cela signifie que l'on souhaite construire un intervalle de confiance à 95% de certitude pour les valeurs des paramètres inconnus on a :

$$\beta_j \in \left[ \hat{\beta}_j \pm t_{n-(p+1), 1-\frac{\alpha}{2}} \cdot \hat{\sigma}_{\beta_j} \right]$$

d'où

$$Pr \left( \hat{\beta}_j - t_{n-(p+1), 1-\frac{\alpha}{2}} \cdot \hat{\sigma}_{\beta_j} \leq \beta_j \leq \hat{\beta}_j + t_{n-(p+1), 1-\frac{\alpha}{2}} \cdot \hat{\sigma}_{\beta_j} \right) = 1 - \alpha$$

## **2.9. ANALYSE DES RESIDUS**

Afin de vérifier entre autre l'ajustement global et la structure de la variance du vecteur , on a souvent recours dans la régression linéaire à certains graphes tels le graphique dressant les résidus standardisés en ordonnée et soit les<sup>^</sup> soit les indices en abscisse, le graphique présentant les racines carrées des valeurs absolues des résidus standardisés en ordonnée et les<sup>^</sup> en abscisse (permet de voir si la dispersion des résidus dépend ou non des valeurs ajustées.), et éventuellement le graphique qq-plot (permet de tester la normalité des résidus). Cette analyse graphique des résidus permet de détecter les différences entre les valeurs observées et les valeurs ajustées (réalisées au moyen de valeurs de et de la droite de régression), ce qui permet de découvrir les faiblesses du modèle vis-à-vis de la moyenne et de la variance mais il ne donne aucune déduction sur la robustesse des estimateurs des par rapport à l'ajout ou la suppression d'une observation qui peut être une valeur aberrante par exemple (il existe d'autres outils dans ce sens).

## **2.10. ANALYSE DE LA VARIANCE (ANOVA)**

### **Définition**

L'analyse de la variance (ANOVA) est une méthode statistique permettant d'évaluer l'influence de différentes sources de variation sur une variable dépendante. Dans le cadre de la régression, l'ANOVA est utilisée pour analyser la contribution individuelle de chaque variable explicative ( $X_1, X_2, \dots, X_p$ ) à l'explication de la variance de  $Y$ .

Permet de vérifier si les coefficients associés aux variables explicatives sont significativement différents de zéro.

## Table ANOVA:

Dans l'ANOVA, la variabilité totale des données est décomposée en trois composantes principales: la somme des carrés, les degrés de liberté (dl) et les carrés moyens, qui sont présentés dans le tableau suivant :

Source de variation	Somme des carrés	ddl	carré moyen	F
régression (expliquée)	$SCE = \sum_{i=1}^n (\hat{y}_i - \bar{y}_n)^2$	p	$\frac{1}{p} \sum_{i=1}^n (\hat{y}_i - \bar{y}_n)^2$	$\frac{SCE/p}{SCR/(n-p-1)}$
Résiduelle	$SCR = \sum_{i=1}^n (y_i - \hat{y}_i)^2$	$n-(p+1)$	$\frac{1}{n-p-1} \sum_{i=1}^n (y_i - \hat{y}_i)^2$	
Totale	$SCT = \sum_{i=1}^n (y_i - \bar{y}_n)^2$	n-1	$\frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y}_n)^2$	

On calcule la valeur de Ftabulée de Fisher, quant à la valeur calculée, elle est notée . La règle de décision du test est : Si  $F > F_{\text{tabulée}}$ , on rejette qui revient à rejeter l'égalité des moyennes.

## Intérêt de l'ANOVA

L'ANOVA permet d'évaluer la qualité d'ajustement du modèle. Si la part de variabilité expliquée (SCE) est significativement plus importante que la part résiduelle (SCR), cela signifie que le modèle est pertinent pour expliquer la variable dépendante. Cet outil est donc crucial pour valider les performances des modèles de régression et guider les prises de décisions basées sur ces modèles.

## **3.CADRE PRATIQUE:**

### **3.1.OVERVIEW SUR LE PROJET**

- **Rapport sur le Dataset**

**Introduction :** Dans le cadre de ce projet, une régression linéaire multiple est appliquée sur un dataset relatif aux coûts d'assurance médicale. Ce dataset contient des informations clés permettant de modéliser et d'estimer les frais médicaux facturés à des assurés en fonction de diverses variables explicatives.

**Description du Dataset :** Ce dataset est composé de 7 colonnes principales :

1. **Age** : L'âge de l'assuré.
2. **Sexe** : Le sexe de l'assuré (homme ou femme).
3. **BMI** (Body Mass Index) : Indice de masse corporelle, une mesure standard utilisée pour évaluer la condition physique.
4. **Children** : Le nombre de personnes à charge.
5. **Smoker** : Indique si l'assuré est fumeur (oui ou non).
6. **Region** : La région géographique où l'assuré réside.
7. **Charges** : Les frais médicaux facturés à l'assuré (variable cible pour les modèles de prédiction).

**Objectif du Projet :** L'objectif principal de ce projet est d'utiliser une régression linéaire multiple pour prédire les coûts d'assurance médicale (également appelés "charges") en fonction des facteurs prédictifs tels que l'âge, le sexe, l'IMC, le nombre d'enfants, le tabagisme et la région de résidence.

Cette analyse vise à :

- Comprendre les relations entre les variables explicatives et la variable cible.
- Identifier les facteurs ayant un impact significatif sur les coûts médicaux.

Le lien de dataset :<https://www.kaggle.com/code/nishxnt/medical-insurance-cost-prediction-python?select=insurance.csv>

- Choix du language Python et les bibliothèques

- Langage Python

Le langage Python a été choisi pour ce projet en raison de sa grande popularité dans le domaine de la science des données, de la machine learning et de la statistique. Python est reconnu pour sa simplicité, sa lisibilité, et ses puissantes bibliothèques dédiées à l'analyse des données, ce qui permet de résoudre efficacement des problèmes complexes sans avoir besoin de coder des solutions complexes à la main. De plus, Python bénéficie d'une large communauté de développeurs et de chercheurs, ce qui permet de bénéficier de nombreuses ressources et d'un support constant.

- Détails sur les bibliothèques utilisées :

Explication des bibliothèques et des méthodes utilisées :

- **Numpy** est une bibliothèque fondamentale pour les calculs numériques en Python. Elle permet de travailler efficacement avec des tableaux multidimensionnels (arrays) et d'exécuter des opérations mathématiques complexes. Dans ce projet, numpy est utilisé pour créer des tableaux avec `numpy.array` et pour calculer la moyenne des données à l'aide de `numpy.mean`.
- **Pandas** est utilisée pour manipuler et analyser les données tabulaires. Avec ses structures de données comme les DataFrames, elle permet un traitement efficace des datasets. Dans ce projet, pandas est utilisé pour charger les données avec `pandas.read_csv` et calculer les statistiques descriptives à l'aide de `pandas.DataFrame.describe`.
- **Matplotlib** est utilisée pour créer des visualisations simples comme les diagrammes de dispersion et les histogrammes. Les relations entre les variables sont visualisées avec `matplotlib.pyplot.scatter`, et les distributions des variables sont analysées avec `matplotlib.pyplot.hist`.

- Seaborn simplifie la création de visualisations statistiques esthétiques. Elle est basée sur matplotlib mais offre des outils avancés. Dans ce projet, seaborn est utilisé pour créer une carte de chaleur des corrélations entre les variables à l'aide de seaborn.heatmap.
- **Scipy** fournit des outils pour les calculs scientifiques et statistiques. Elle est utilisée ici pour effectuer une régression linéaire et obtenir des statistiques associées telles que le coefficient de corrélation avec scipy.stats.linregress.
- **Statsmodels** est une bibliothèque puissante pour les analyses statistiques. Elle fournit des outils pour effectuer des régressions et interpréter les résultats en détail. Dans ce projet, statsmodels est utilisé pour implémenter des modèles de régression linéaire avec statsmodels.api.OLS et pour estimer les paramètres du modèle avec fit.
- **Scikit-learn** est incontournable pour la machine learning. Elle fournit des outils pour la préparation des données, l'entraînement et l'évaluation des modèles. Dans ce projet, scikit-learn est utilisé pour créer un modèle de régression linéaire multiple avec sklearn.linear\_model.LinearRegression, entraîner le modèle avec fit et prédire les valeurs basées sur les données d'entrée avec predict.
- **Warnings** est utilisée pour gérer ou ignorer les avertissements Python afin de garantir une exécution propre du code sans interruptions inutiles. Les avertissements sont supprimés avec warnings.filterwarnings("ignore").

## 3.2. LE CODE

### 1. Importation des données

Les bibliothèques :

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

Chargement des données:

```
file_path = 'insurance.csv'
data = pd.read_csv(file_path)
```

Dimension du dataset:

```
info_df = pd.DataFrame({
    'Nombre de lignes': [data.shape[0]],
    'Nombre de colonnes': [data.shape[1]]
})
info_df
```

	Nombre de lignes	Nombre de colonnes
0	1338	7

Affichage des informations sur les colonnes, les types de données et quelques informations

```
print(data.info())
```

```
Informations sur le dataset :
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          -----          ----- 
 0   age         1338 non-null   int64  
 1   sex         1338 non-null   object  
 2   bmi         1338 non-null   float64 
 3   children    1338 non-null   int64  
 4   smoker      1338 non-null   object  
 5   region      1338 non-null   object  
 6   charges     1338 non-null   float64 
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
None
```

## 2. Analyse exploratoire des données

### Aperçu des Données:

Exemple des premières lignes du dataset :

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

En examinant les premières lignes du dataset, il est évident que chaque colonne joue un rôle unique dans l'ensemble de données.

Par exemple :

**age** peut influencer directement les frais d'assurance en raison des risques accusés avec l'âge.

**smoker** est une variable catégorique cruciale car les fumeurs ont des frais nettement plus élevés.

**bmi** reflète l'état de santé de l'individu, ce qui peut avoir un impact significatif sur les charges.

**region** permet d'examiner les variations géographiques potentielles dans les coûts.

### Statistiques descriptives :

Statistiques descriptives :				
	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

## Analyse Statistique

Les statistiques descriptives mettent en évidence les caractéristiques principales du dataset :

- Âge moyen : 39 ans, avec un écart-type de 14 ans, couvrant une population de 18 à 64 ans.
- IMC moyen : 30.66, indiquant une tendance vers le surpoids pour la plupart des individus.
- Frais moyens d'assurance : 13,270 USD, avec une forte variabilité (min : 1,121 USD, max : 63,770 USD).

## Relations entre les Variables

- Fumeur vs Charges : Les fumeurs paient en moyenne des frais bien plus élevés que les non-fumeurs.
- IMC vs Charges : Une corrélation positive, surtout chez les fumeurs.
- Age : Les frais augmentent généralement avec l'âge.

En analysant ces statistiques, il est possible d'identifier les tendances générales et de planifier les prochaines étapes d'analyse et de modélisation de manière structurée.

## 3. Nettoyage du dataset

### Vérifier les valeurs manquantes:

```
: print(data.isnull().sum())
```

age	0
sex	0
bmi	0
children	0
smoker	0
region	0
charges	0

dtype: int64

Le dataset ne contient aucune valeur manquante

### Verification et suppression des doublons:

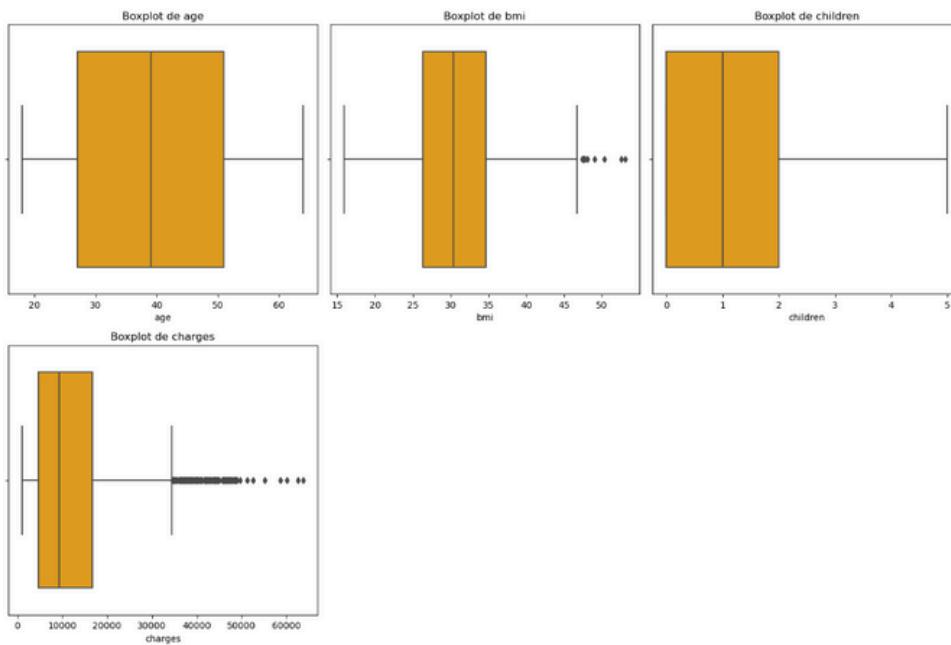
```
: duplicated_rows = data.duplicated().sum()  
# Afficher le nombre de doublons  
print(f"Nombre de lignes dupliquées : {duplicated_rows}")
```

Nombre de lignes dupliquées : 1

```
data = data.drop_duplicates()
```

Aucune valeur doublon n'a été détecté maintenant dans l'ensemble des données

## Verification des valeurs aberrantes :



L'identification des valeurs abérantes est une étape cruciale pour garantir la qualité du modèle et éviter les biais dans les résultats. Une valeur abérante peut se manifester par une donnée extrême dans une ou plusieurs variables.

Des boîtes à moustaches (boxplots) ont été générées pour identifier les valeurs abérantes dans les variables charges, bmi, children et age. Les résultats montrent :

**age** : Aucune valeur aberrante détectée. Les données d'âge sont bien réparties dans l'intervalle [20, 60].

**bmi** : Quelques valeurs aberrantes sont présentes du côté supérieur ( $> 40$ ). Cela représente des individus avec un indice de masse corporelle particulièrement élevé, ce qui pourrait indiquer des cas d'obésité extrême.

**children** : Aucune valeur aberrante détectée. Les données sur le nombre d'enfants se situent dans la plage attendue [0, 5].

**Charges (Coût des assurances)** : De nombreuses valeurs aberrantes sont présentes dans les charges élevées ( $> 30,000$ ). Ces valeurs peuvent représenter des cas exceptionnels, comme des individus nécessitant des soins coûteux (ex. : fumeurs ou maladies graves)

## Suppression des valeurs aberrantes :

```
for col in numerical_columns:  
    Q1 = data[col].quantile(0.25)  
    Q3 = data[col].quantile(0.75)  
    IQR = Q3 - Q1  
    lower_bound = Q1 - 1.5 * IQR  
    upper_bound = Q3 + 1.5 * IQR  
    data = data[(data[col] >= lower_bound) & (data[col] <= upper_bound)]  
  
print("\nTaille des données après suppression des valeurs aberrantes :", data.shape)  
taille des données après suppression des valeurs aberrantes : (1190, 7)
```

## Encodage des variables catégoriques:

```
from sklearn.preprocessing import LabelEncoder  
  
# Appliquer LabelEncoder sur chaque colonne catégorielle  
encoder = LabelEncoder()  
  
for column in ['sex', 'smoker', 'region']: # Colonnes catégorielles  
    data[column] = encoder.fit_transform(data[column])
```

L'encodage des variables catégoriques est une étape essentielle pour permettre leur intégration dans un modèle de régression linéaire, qui ne traite que des variables numériques. Voici les étapes effectuées :

Variables encodées :

**sex** : Transformée en deux colonnes indicatrices (sex\_male et sex\_female), avec une valeur binaire pour indiquer le sexe.

**smoker** : Encodée en deux colonnes (smoker\_yes et smoker\_no), la colonne smoker\_yes étant utilisée comme indicatrice clé.

**region** : Transformée en quatre colonnes indicatrices (region\_northeast, region\_northwest, region\_southeast, region\_southwest).

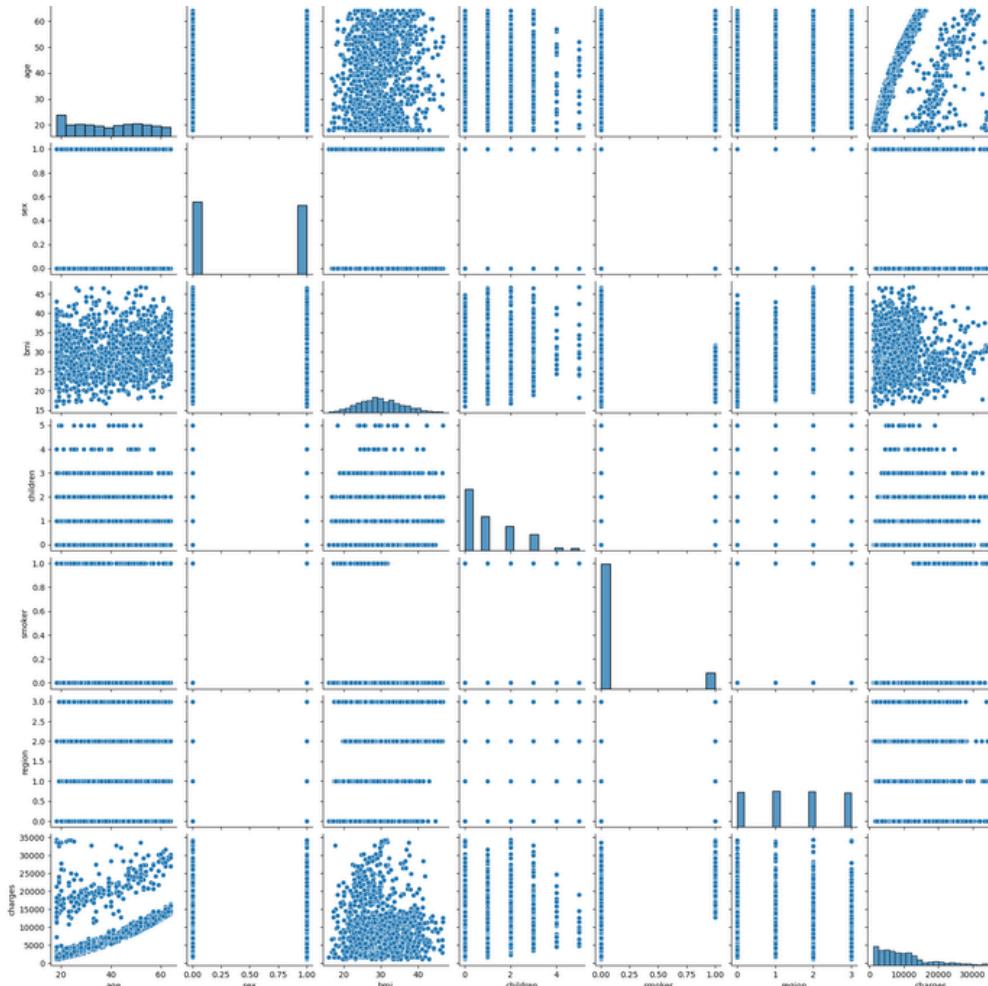
## Outils utilisés :

La méthode OneHotEncoder de sklearn.preprocessing a été utilisée pour automatiser le processus et éviter les biais d'ordonnancement dans les variables catégoriques.

### 3. Visualisation des données

#### Scatter plot pour chaque paire de variables

```
# Créez un scatter plot pour chaque paire de variables
sns.pairplot(data)
plt.show()
```

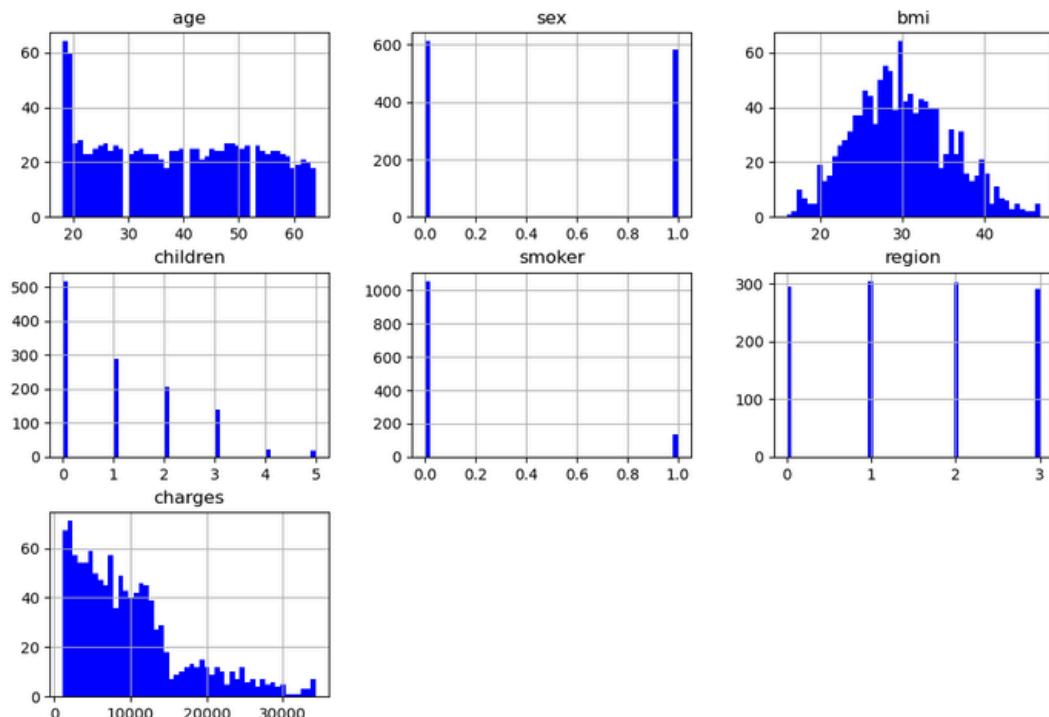


Le graphique pairplot fournit une vue d'ensemble des relations entre les variables du dataset. Les histogrammes sur la diagonale montrent la distribution de chaque variable. Par exemple, age semble avoir une distribution assez uniforme, tandis que charges est asymétrique, avec une majorité de valeurs faibles et quelques valeurs très élevées. Concernant les relations entre variables, on observe que les charges augmentent avec l'âge, ce qui peut être expliqué par une augmentation des besoins médicaux avec le vieillissement. De même, une corrélation positive apparaît entre bmi et charges, indiquant que les personnes ayant un indice de masse corporelle élevé ont tendance à avoir des charges médicales plus importantes.

Les variables catégoriques, comme `smoker`, montrent des différences marquées : les charges sont significativement plus élevées pour les fumeurs par rapport aux non-fumeurs, soulignant l'impact du tabagisme sur les coûts. En revanche, les variables `sex` et `region` ne semblent pas influencer les charges de manière notable. Enfin, certains clusters visibles dans les scatterplots, notamment liés à `smoker`, confirment la segmentation claire des données selon cette variable. Ces observations permettent d'identifier les variables les plus importantes pour expliquer ou prédire les charges médicales.

## Histogramme pour chaque variable numérique

```
data.hist(bins=50 , figsize=(12,8), color='blue')
plt.show()
```



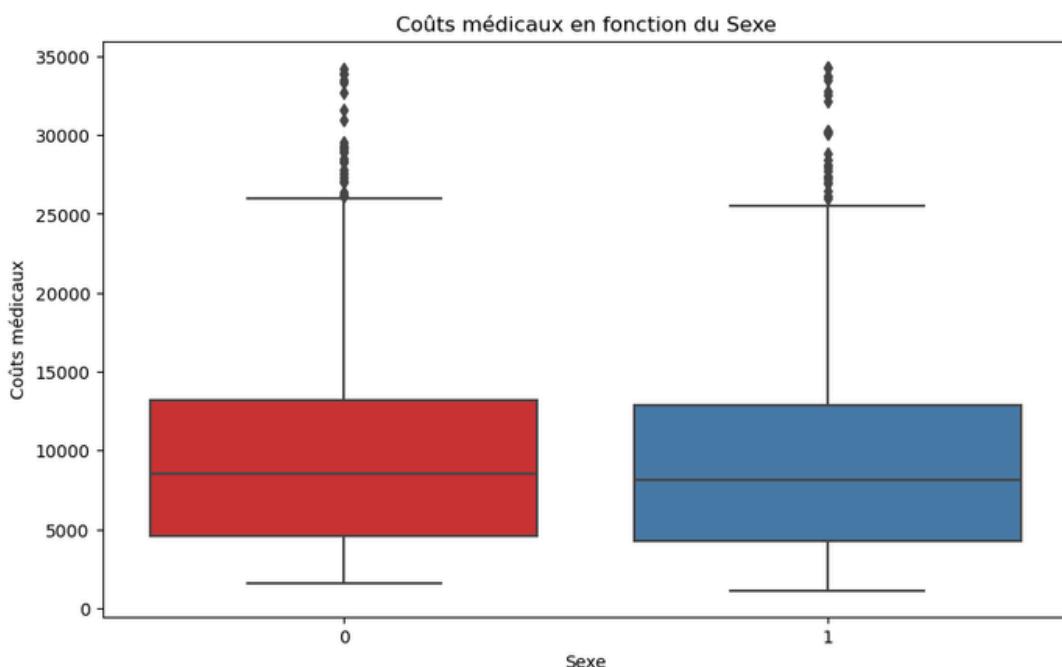
Les histogrammes des différentes variables du dataset offrent une vue d'ensemble de leurs distributions. La variable `age` est uniformément répartie entre 20 et 60 ans, indiquant un échantillonnage équilibré des âges. La variable `sex`, binaire, montre une répartition quasi égale entre les deux catégories (hommes et femmes). Concernant `bmi`, la distribution est centrée autour de 30, avec une légère asymétrie vers les valeurs élevées, ce qui est attendu dans une population adulte. Pour `children`, la majorité des individus ont entre 0 et 3 enfants, tandis que les familles avec plus d'enfants sont rares. La variable `smoker`

révèle une nette majorité de non-fumeurs, tandis que les fumeurs représentent une minorité significative. La variable `region`, catégorielle, montre une répartition uniforme entre quatre régions, ce qui suggère l'absence de biais géographique. Enfin, la variable `charges`, représentant les dépenses médicales, est fortement asymétrique : la majorité des individus ont des charges faibles, mais quelques cas présentent des coûts très élevés, probablement dus à des situations médicales exceptionnelles. Cette analyse des distributions permet d'identifier les caractéristiques clés des données et d'envisager les relations possibles entre ces variables.

## Boxplots pour les variables numériques par rapport aux variables catégorielles

```
# Vérification des colonnes et conversion en catégorie
print(data.columns)
data['sex'] = data['sex'].astype('category')

# Création du graphique boxplot
plt.figure(figsize=(10, 6))
sns.boxplot(x='sex', y='charges', data=data, palette='Set1')
plt.title('Coûts médicaux en fonction du Sexe')
plt.xlabel('Sexe')
plt.ylabel('Coûts médicaux')
plt.show()
```



Ce graphique présenté est un boxplot qui illustre la répartition des coûts médicaux en fonction du sexe. Les données utilisées pour cette visualisation ont d'abord été préparées, notamment en convertissant la

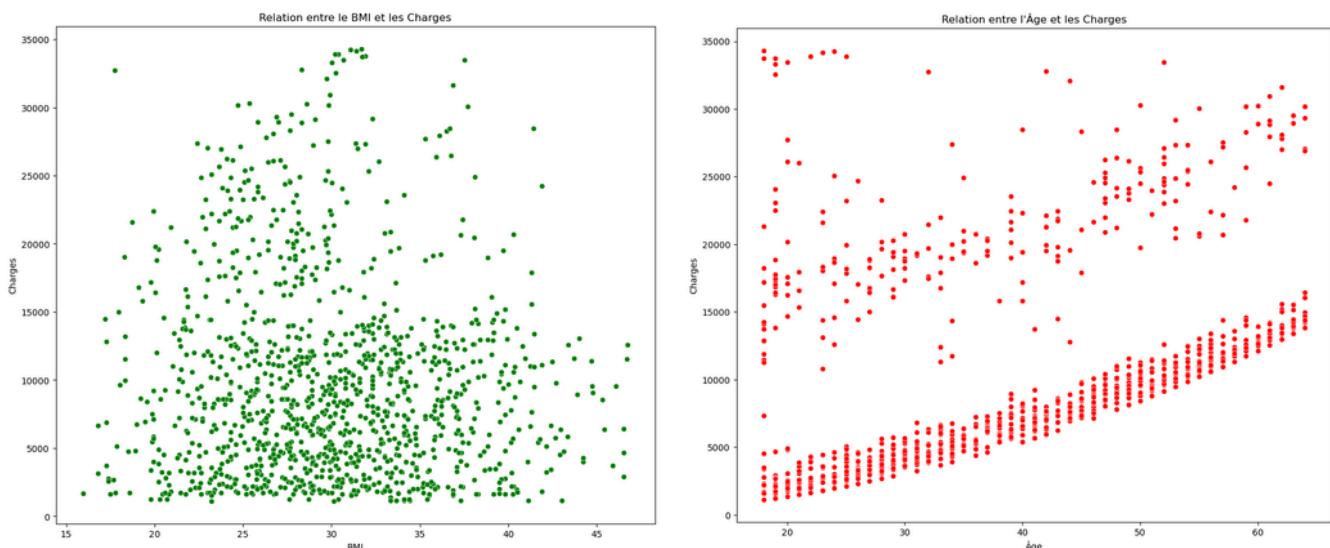
colonne "sex" en une catégorie, ce qui est indiqué par le code `data['sex'] = data['sex'].astype('category')`. Le boxplot est créé en utilisant la bibliothèque Seaborn avec l'axe des abscisses représentant le sexe (codé sous forme de 0 et 1) et l'axe des ordonnées représentant les coûts médicaux.

Chaque boîte montre la médiane des coûts médicaux, les quartiles, ainsi que les valeurs aberrantes sous forme de points au-delà des moustaches. Le graphique met en évidence une différence potentielle dans la distribution des coûts médicaux entre les deux sexes. Une palette de couleurs est utilisée pour différencier les catégories. Le titre du graphique, ainsi que les étiquettes des axes, ont été ajoutés pour clarifier les informations.

## Nuages de points (scatter plots) pour les relations entre les variables numériques

```
# Nuage de points pour les variables numériques
plt.figure(figsize=(12, 10))
sns.scatterplot(x='bmi', y='charges', data=data, color='green')
plt.title('Relation entre le BMI et les Charges')
plt.xlabel('BMI')
plt.ylabel('Charges')
plt.show()

plt.figure(figsize=(12, 10))
sns.scatterplot(x='age', y='charges', data=data, color='red')
plt.title('Relation entre l\'Âge et les Charges')
plt.xlabel('Âge')
plt.ylabel('Charges')
plt.show()
```



- Relation entre le BMI et les Charges (graphique de gauche) :

Le nuage de points utilise une couleur verte pour représenter les données. L'axe des abscisses correspond au BMI, tandis que l'axe des ordonnées

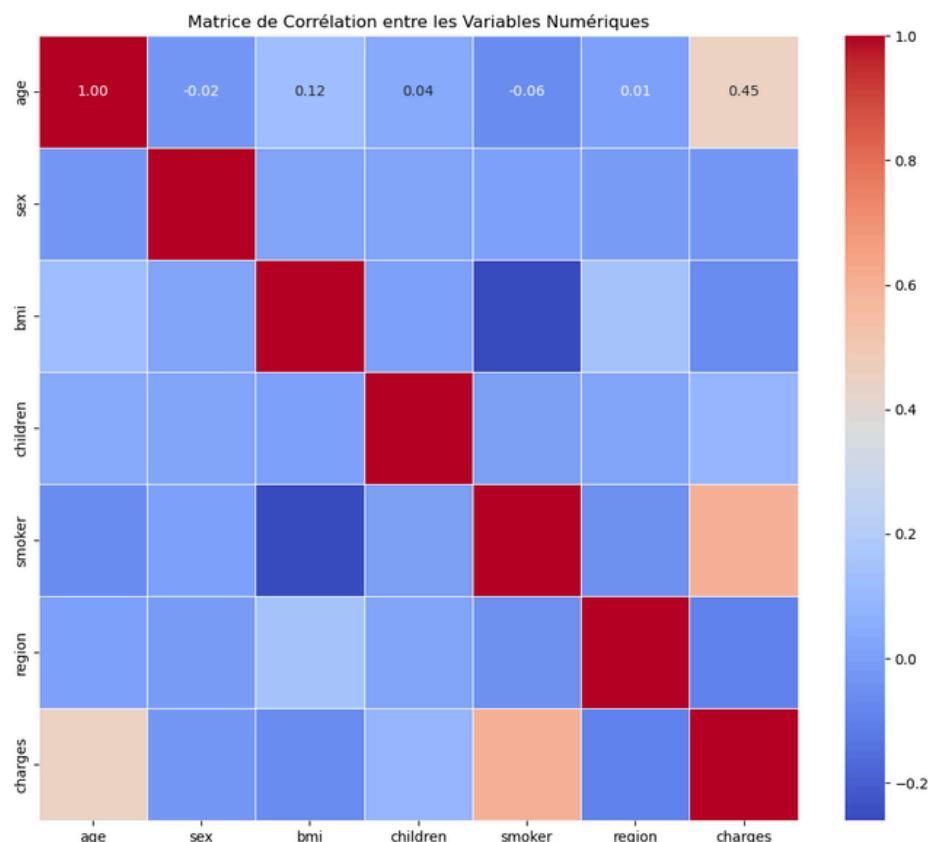
représente les charges. Ce graphique permet d'observer si une tendance existe entre l'IMC et les coûts médicaux. On peut noter une dispersion importante, mais certains points suggèrent que des coûts médicaux élevés sont associés à des BMI plus élevés.

- Relation entre l'Âge et les Charges (graphique de droite) :

Le nuage de points utilise une couleur rouge. L'axe des abscisses représente l'âge, et l'axe des ordonnées montre les coûts médicaux. Contrairement au BMI, une tendance plus claire se dessine ici, où les coûts médicaux semblent augmenter avec l'âge. Les points forment une structure ascendante, indiquant potentiellement une corrélation positive entre l'âge et les charges.

## la matrice de corrélation

```
: # Matrice de corrélation
plt.figure(figsize=(12, 10))
correlation_matrix = data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Matrice de Corrélation entre les Variables Numériques')
plt.show()
```

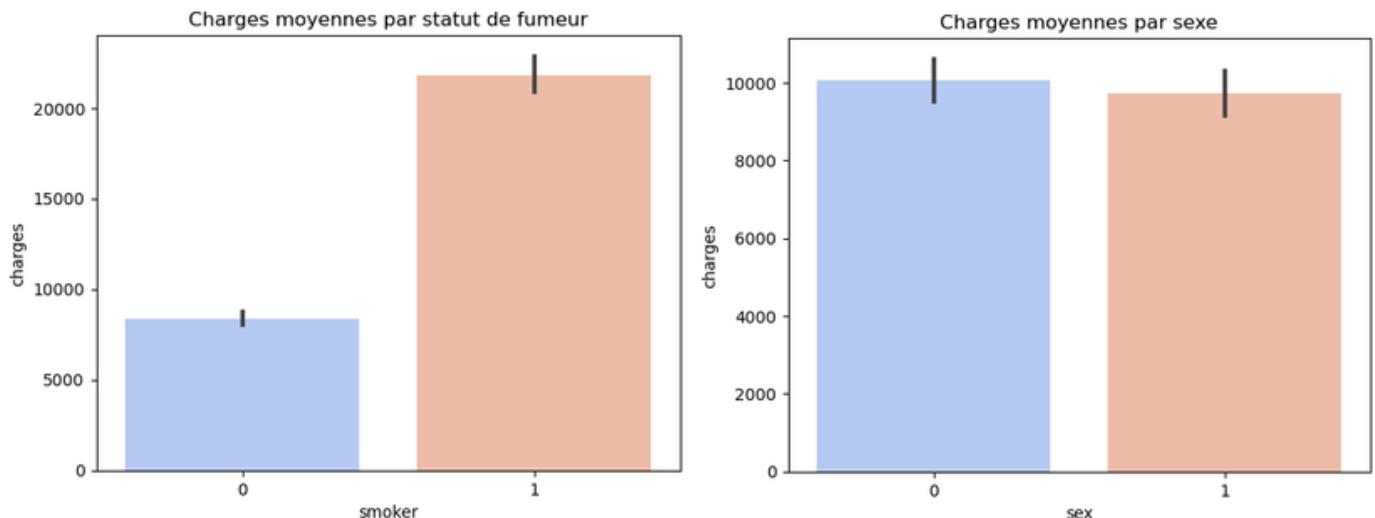


La matrice de corrélation met en évidence les relations entre les variables numériques. On observe que le fait de fumer ("smoker") est fortement corrélé aux coûts médicaux ("charges"), tandis que l'âge et l'IMC ("bmi") montrent des corrélations modérées avec les charges. Les autres variables, comme le sexe, le nombre d'enfants et la région, présentent des corrélations faibles, suggérant un impact limité sur les coûts médicaux. Ce graphique identifie les facteurs les plus influents sur les charges médicales.

## Visualisation des variables catégorielles avec des pourcentages

```
20]: # Pourcentage de fumeurs/non-fumeurs par rapport aux charges
sns.barplot(x='smoker', y='charges', data=data, estimator=np.mean, palette='coolwarm')
plt.title('Charges moyennes par statut de fumeur')
plt.show()

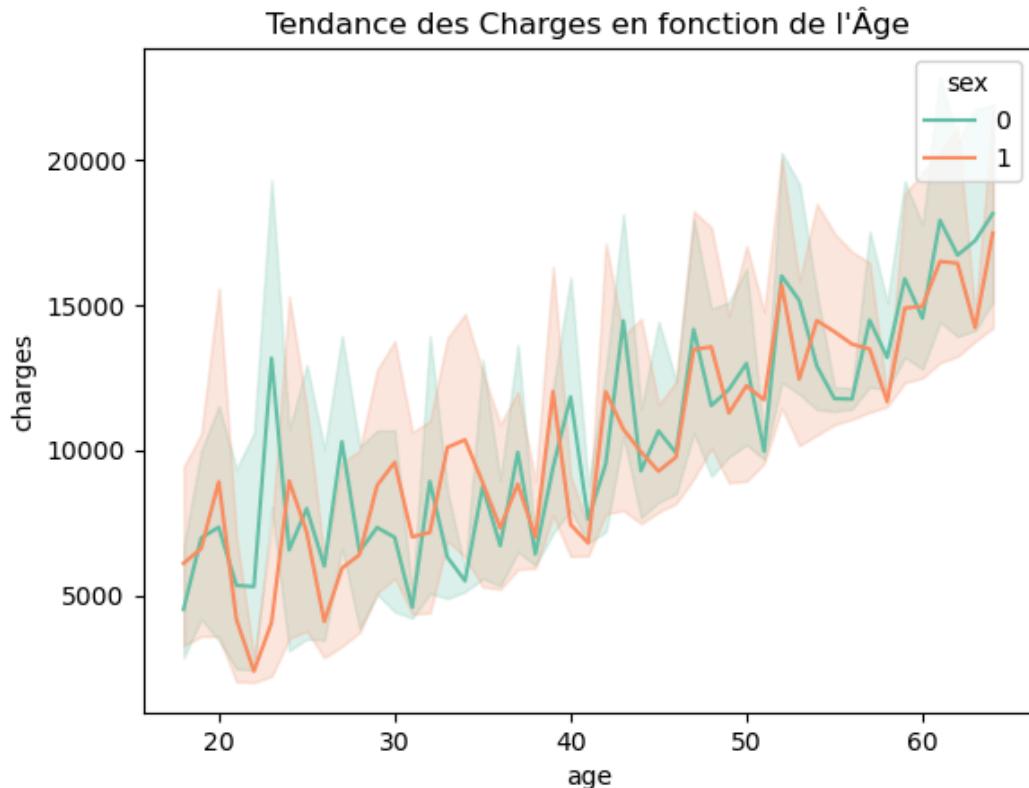
# Pourcentage par sexe
sns.barplot(x='sex', y='charges', data=data, estimator=np.mean, palette='coolwarm')
plt.title('Charges moyennes par sexe')
plt.show()
```



Les graphiques comparent les charges moyennes selon deux critères : le statut de fumeur et le sexe. Le premier graphique montre que les fumeurs ont des charges nettement plus élevées que les non-fumeurs. Le second révèle une légère différence entre les sexes, les hommes ayant des charges moyennes un peu plus élevées que les femmes. Ces visualisations mettent en évidence des tendances intéressantes et aident à mieux comprendre comment ces facteurs influencent les coûts, avec des barres d'erreur pour refléter la variation des données.

## Graphiques de tendances

```
|: # Relation entre l'âge et Les charges
sns.lineplot(x='age', y='charges', data=data, hue='sex', markers=["o", "s"], palette='Set2')
plt.title('Tendance des Charges en fonction de l'Âge')
plt.show()
```



Ce graphique illustre l'évolution des charges en fonction de l'âge, en distinguant les individus selon leur sexe (0 pour les femmes et 1 pour les hommes). On observe une tendance générale à la hausse des charges avec l'âge, quel que soit le sexe, bien que la progression semble légèrement plus marquée pour les hommes. Les zones ombrées autour des courbes représentent les intervalles de confiance, indiquant la variabilité des données. Cette visualisation met en évidence que les charges augmentent avec l'âge, tout en montrant des différences subtiles entre les sexes.

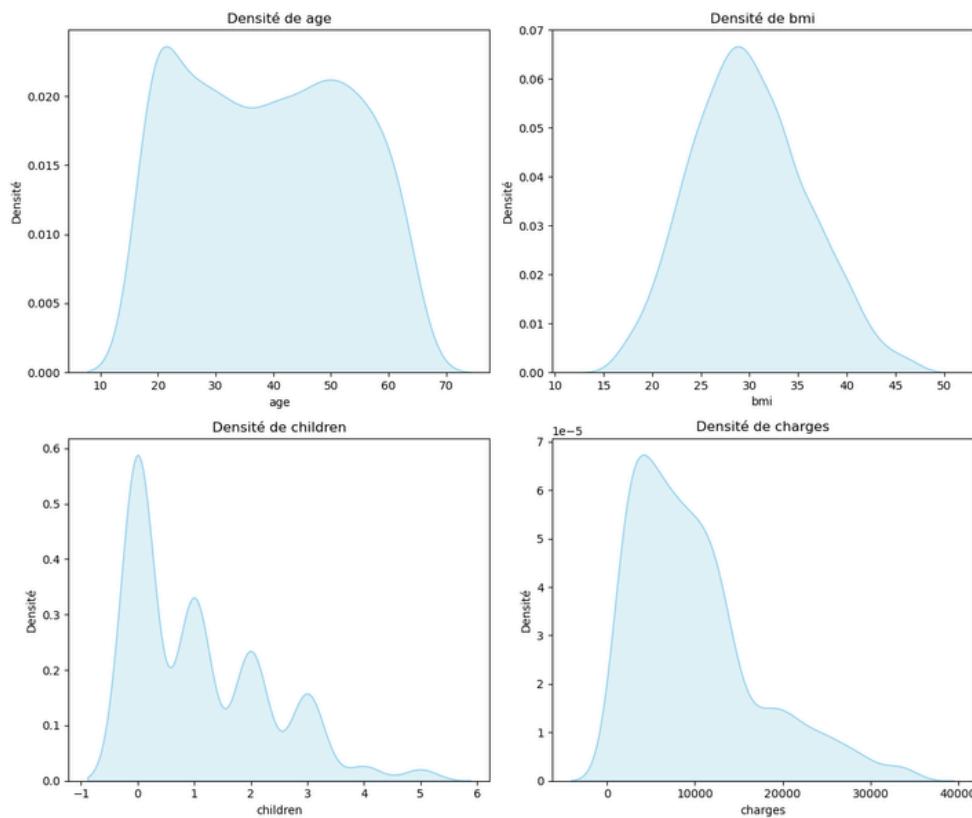
## Diagrammes de densité

```
|: # Diagrammes de densité pour toutes les variables
plt.figure(figsize=(12, 10))

# Liste des variables numériques pour les diagrammes de densité
numerical_columns = ['age', 'bmi', 'children', 'charges']

for i, col in enumerate(numerical_columns, 1):
    plt.subplot(2, 2, i)
    sns.kdeplot(data[col], shade=True, color='skyblue')
    plt.title(f'Densité de {col}')
    plt.xlabel(col)
    plt.ylabel('Densité')

plt.tight_layout()
plt.show()
```



Les diagrammes de densité montrent la répartition des variables numériques.

1. Âge : La majorité des individus sont âgés entre 20 et 50 ans.
2. BMI (Indice de Masse Corporelle) : La distribution est centrée autour de 30, indiquant que la plupart des individus ont un IMC dans la catégorie "surpoids".
3. Nombre d'enfants : La plupart ont entre 0 et 2 enfants, avec une décroissance au-delà.
4. Charges : Les charges sont asymétriques, concentrées sous 10 000, avec des valeurs plus rares au-delà de 20 000.

Ces distributions aident à comprendre la structure des données et à identifier les tendances générales.

## 4. Regression sur chaque variable:

### Ajout de la constante dans le modèle

```
from scipy.stats import f
import statsmodels.api as sm
from statsmodels.graphics.gofplots import qqplot
from statsmodels.stats.anova import anova_lm
def lm(x):
    data['const'] = 1 # Ajouter une constante pour le modèle
```

La constante est ajoutée au modèle pour représenter l'ordonnée à l'origine (intercept) de la régression. Cela permet de modéliser la relation entre la variable indépendante et la variable dépendante en tenant compte d'un décalage constant.

### Définition des variables dépendantes et indépendantes

```
# Définir la variable dépendante et les variables indépendantes
Y = data['charges'] # Variable cible
X = data[['const', 'x']] # Variable explicative + constante
```

Variable dépendante (Y) : C'est la variable que tu cherches à prédire. Dans ce cas, il s'agit de charges.

Variable indépendante (X) : Il s'agit de la variable explicative, ici la colonne x, à laquelle tu associes une constante.

### Ajustement du modèle linéaire (OLS)

```
# Ajuster le modèle linéaire
lm_model = sm.OLS(Y, X).fit()
```

Le modèle de régression linéaire est ajusté à l'aide de la méthode des moindres carrés ordinaires (OLS) via `sm.OLS()`. Le modèle est ensuite ajusté avec `.fit()`.

### Affichage des résultats du modèle

```
print(lm_model.summary())
```

Une fois le modèle ajusté, tu affiches un résumé des résultats. Le résumé contient des informations telles que les coefficients estimés, les erreurs standard, les valeurs p, le  $R^2$ , etc.

## QQ Plot des résidus

```
# Calculer et afficher le QQ plot des résidus
residuals = lm_model.resid
qqplot(residuals, line='s', color='blue')
plt.title(f"QQ Plot des résidus pour la variable : {x}")
plt.show()
```

Le QQ plot permet de visualiser la normalité des résidus du modèle. Si les résidus suivent une distribution normale, ils devraient se situer le long de la ligne droite sur le graphique.

## Calcul des métriques du modèle

```
R2_model = lm_model.rsquared
R2_adj_model = lm_model.rsquared_adj
Fcalc_model = lm_model.fvalue

n = data.shape[0] # Nombre d'observations
p = 1 # Nombre de prédicteurs (excluant la constante)
Ftbl_model = f.isf(0.05, p, n - (p + 1)) # Valeur critique de F
```

- $R^2$  : Coefficient de détermination qui mesure la proportion de la variance de la variable cible expliquée par le modèle. Un  $R^2$  plus élevé indique que le modèle s'ajuste mieux aux données.
- $R^2$  ajusté : Prend en compte le nombre de variables explicatives, corrigeant ainsi l'effet du nombre de prédicteurs sur l'évaluation du modèle.
- Valeur de F calculée : Une statistique qui permet de tester l'hypothèse selon laquelle tous les coefficients des variables explicatives sont égaux à zéro (c'est-à-dire qu'aucune des variables n'explique la variation de la variable cible).
- Valeur critique de F ( $F_{tbl}$ ) : La valeur critique de la statistique F à partir de la distribution de Fisher, utilisée pour tester l'hypothèse nulle selon laquelle le modèle n'a pas de pouvoir explicatif.

## Retour des métriques dans un DataFrame

```
return pd.DataFrame(
    [R2_model * 100, R2_adj_model * 100, Fcalc_model, Ftbl_model],
    index=['R2', 'R2_adj', 'Fcalc', 'Ftbl_model'],
    columns=[x]
)
```

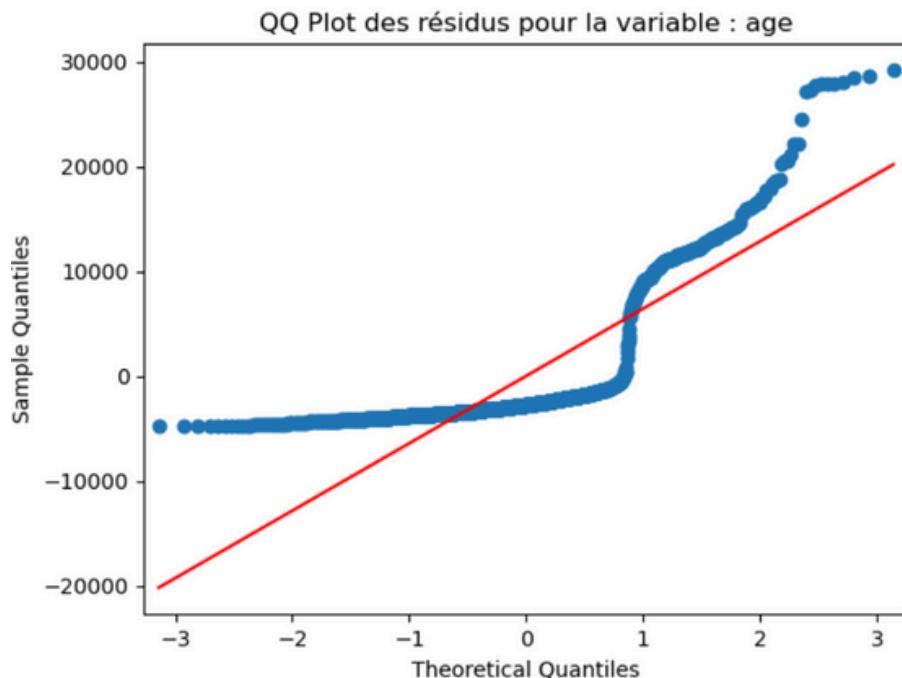
## Regression sur 'age'

```
: result = lm('age')
print(result)

Modèle pour la variable indépendante : age
OLS Regression Results
=====
Dep. Variable: charges    R-squared:      0.201
Model:          OLS         Adj. R-squared:   0.200
Method:        Least Squares  F-statistic:     298.4
Date:       Sun, 19 Jan 2025  Prob (F-statistic): 8.03e-60
Time:        17:28:28        Log-Likelihood: -12122.
No. Observations: 1190        AIC:             2.425e+04
Df Residuals: 1188        BIC:             2.426e+04
Df Model:      1
Covariance Type: nonrobust
=====
            coef    std err      t      P>|t|      [0.025      0.975]
-----
const    964.2833   550.260     1.752     0.080    -115.306    2043.873
age      229.1643   13.266    17.274     0.000     203.136    255.192
=====
Omnibus:        450.706  Durbin-Watson:   2.017
Prob(Omnibus):  0.000   Jarque-Bera (JB): 1314.492
Skew:           1.980   Prob(JB):      3.65e-286
Kurtosis:       6.292   Cond. No.       123.
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



	age
R2	20.075088
R2_adj	20.007811
Fcalc	298.395122
Ftbl_model	3.849298

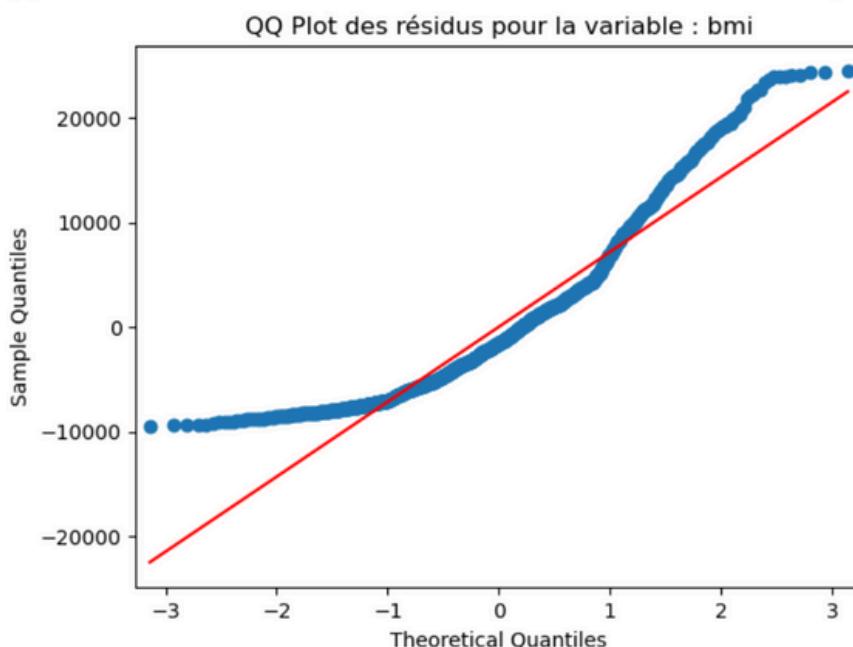
On a Fcalc >>> Ftbl\_model donc on rejette au niveau de risque 5% l'hypothèse H0 d'où le modèle est globalement significatif.

## Regression sur 'bmi'

```
Modèle pour la variable indépendante : bmi
OLS Regression Results
=====
Dep. Variable: charges    R-squared:      0.004
Model:          OLS         Adj. R-squared:   0.003
Method:        Least Squares  F-statistic:    4.951
Date:       Sun, 19 Jan 2025  Prob (F-statistic): 0.0263
Time:        17:28:28       Log-Likelihood: -12253.
No. Observations: 1190      AIC:           2.451e+04
Df Residuals:   1188      BIC:           2.452e+04
Df Model:        1
Covariance Type: nonrobust
=====
            coef    std err      t      P>|t|      [0.025      0.975]
-----
const     1.227e+04  1082.740    11.334    0.000    1.01e+04  1.44e+04
bmi      -78.8268   35.427    -2.225    0.026   -148.334   -9.320
=====
Omnibus:             196.743  Durbin-Watson:    2.033
Prob(Omnibus):       0.000   Jarque-Bera (JB): 302.908
Skew:                 1.140   Prob(JB):       1.68e-66
Kurtosis:              3.955   Cond. No.        159.
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



	bmi
R2	0.415001
R2_adj	0.331175
Fcalc	4.950753
Ftbl_model	3.849298

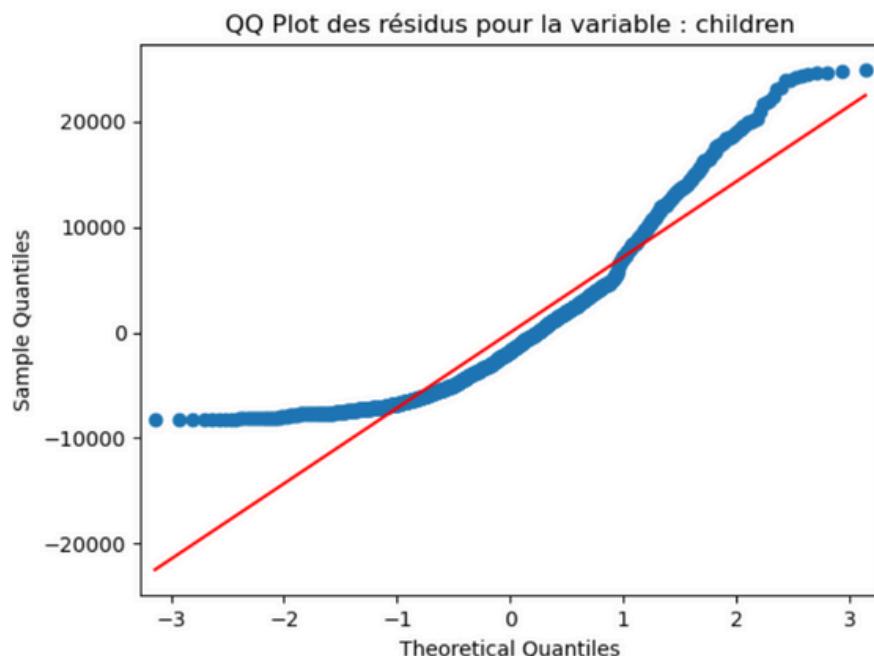
On a  $F_{\text{calc}} > F_{\text{tbl\_model}}$  donc on rejette au niveau de risque 5% l'hypothèse  $H_0$  d'où le modèle est globalement significatif

## Regression sur 'children'

```
Modèle pour la variable indépendante : children
OLS Regression Results
=====
Dep. Variable: charges    R-squared:      0.008
Model:          OLS         Adj. R-squared:   0.007
Method:        Least Squares F-statistic:   9.335
Date:       Sun, 19 Jan 2025 Prob (F-statistic): 0.00230
Time:        17:28:29    Log-Likelihood: -12251.
No. Observations: 1190    AIC:           2.451e+04
Df Residuals: 1188    BIC:           2.452e+04
Df Model:      1
Covariance Type: nonrobust
=====
            coef    std err      t      P>|t|      [0.025      0.975]
-----
const     9340.4374  278.586   33.528    0.000    8793.863    9887.012
children  521.5869  170.715    3.055    0.002    186.651    856.523
=====
Omnibus:             211.964 Durbin-Watson:      2.038
Prob(Omnibus):       0.000  Jarque-Bera (JB): 336.595
Skew:                 1.201  Prob(JB):       8.12e-74
Kurtosis:              4.011  Cond. No.       2.63
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



children	
R2	0.779642
R2_adj	0.696123
Fcalc	9.334929
Ftbl_model	3.849298

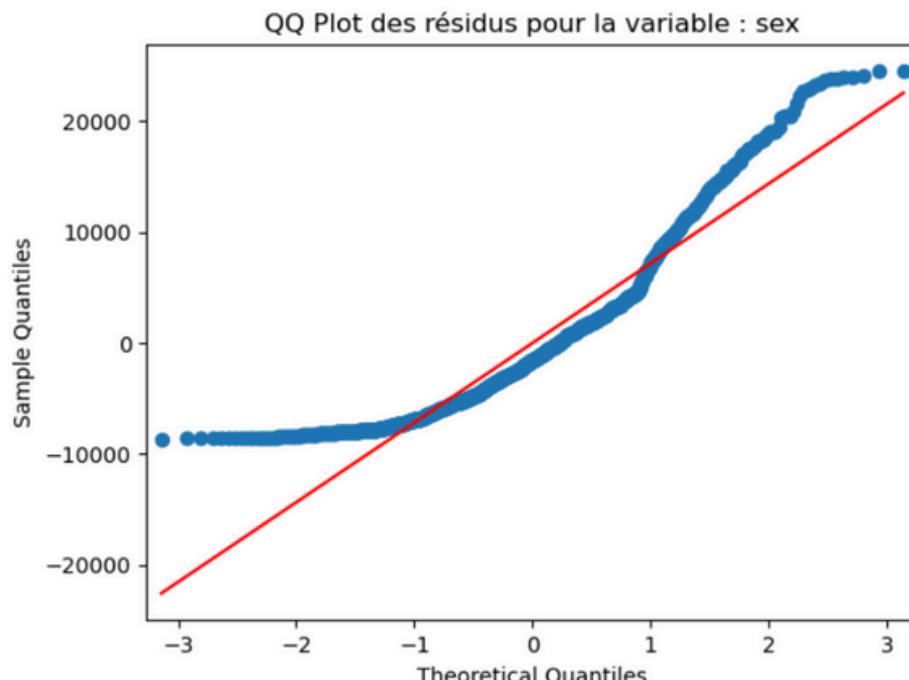
On a  $F_{\text{calc}} > F_{\text{tbl\_model}}$  donc on rejette au niveau de risque 5% l'hypothèse  $H_0$  d'où le modèle est globalement significatif

## Regression sur 'sex'

```
Modèle pour la variable indépendante : sex
OLS Regression Results
=====
Dep. Variable: charges R-squared:      0.000
Model:          OLS   Adj. R-squared:   -0.000
Method:         Least Squares F-statistic:     0.5713
Date:           Sun, 19 Jan 2025 Prob (F-statistic): 0.450
Time:           17:28:29 Log-Likelihood:    -12255.
No. Observations: 1190   AIC:            2.451e+04
Df Residuals:   1188   BIC:            2.452e+04
Df Model:        1
Covariance Type: nonrobust
=====
            coef    std err      t      P>|t|      [0.025      0.975]
-----
const    1.006e+04   291.022    34.573    0.000    9490.645    1.06e+04
sex     -315.0862   416.855   -0.756    0.450   -1132.939    502.767
=====
Omnibus:             201.642 Durbin-Watson:       2.037
Prob(Omnibus):       0.000  Jarque-Bera (JB): 313.143
Skew:                 1.162  Prob(JB):        1.00e-68
Kurtosis:              3.956  Cond. No.        2.59
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



sex	
R2	0.048069
R2_adj	-0.036066
Fcalc	0.571333
Ftbl_model	3.849298

On a  $F_{\text{calc}} < F_{\text{tbl\_model}}$  donc on accepte au niveau de risque 5% l'hypothèse  $H_0$  d'où le modèle n'est pas significatif

## Regression sur 'smoker'

Modèle pour la variable indépendante : smoker

OLS Regression Results

Dep. Variable:	charges	R-squared:	0.355
Model:	OLS	Adj. R-squared:	0.355
Method:	Least Squares	F-statistic:	655.3
Date:	Sun, 19 Jan 2025	Prob (F-statistic):	1.87e-115
Time:	17:28:30	Log-Likelihood:	-11994.
No. Observations:	1190	AIC:	2.399e+04
Df Residuals:	1188	BIC:	2.400e+04
Df Model:	1		
Covariance Type:	nonrobust		

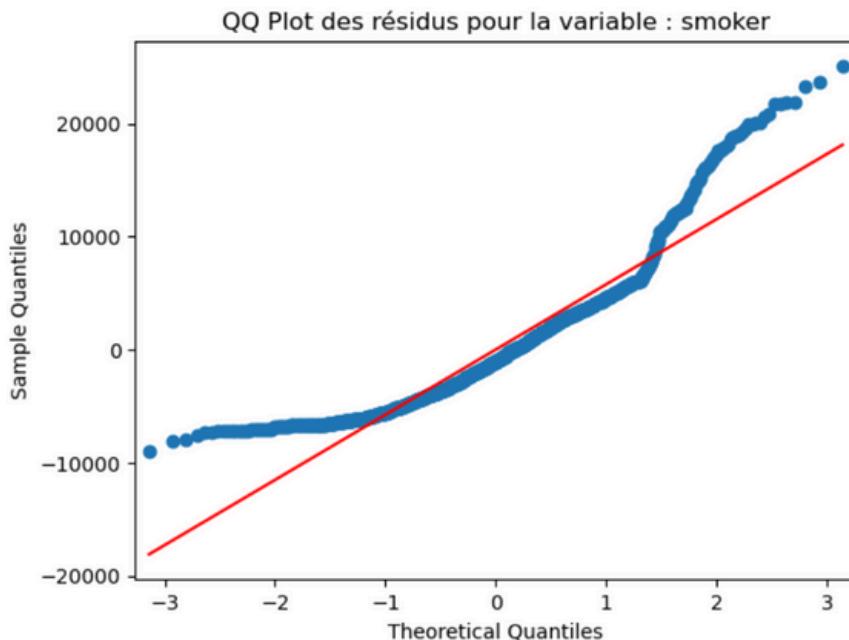
	coef	std err	t	P> t	[0.025	0.975]
const	8369.5657	177.782	47.078	0.000	8020.763	8718.368
smoker	1.346e+04	525.888	25.598	0.000	1.24e+04	1.45e+04

Omnibus:	291.017	Durbin-Watson:	2.051
Prob(Omnibus):	0.000	Jarque-Bera (JB):	633.513
Skew:	1.363	Prob(JB):	2.72e-138
Kurtosis:	5.311	Cond. No.	3.19

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



smoker
R2
35.549015
R2_adj
35.494763
Fcalc
655.261201
Ftbl_model
3.849298

On a Fcalc >>> Ftbl\_model donc on rejette au niveau de risque 5% l'hypothèse H0 d'ou le modèle est globalement significatif

## Regression sur 'region'

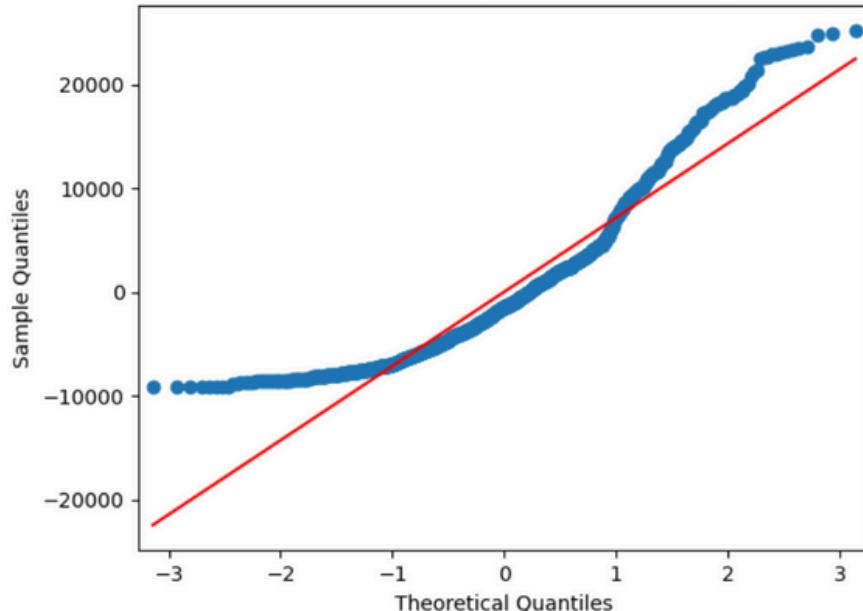
Modèle pour la variable indépendante : region  
OLS Regression Results

Dep. Variable:	charges	R-squared:	0.009
Model:	OLS	Adj. R-squared:	0.009
Method:	Least Squares	F-statistic:	11.29
Date:	Sun, 19 Jan 2025	Prob (F-statistic):	0.000804
Time:	17:28:31	Log-Likelihood:	-12250.
No. Observations:	1190	AIC:	2.450e+04
Df Residuals:	1188	BIC:	2.451e+04
Df Model:	1		
Covariance Type:	nonrobust		

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

QQ Plot des résidus pour la variable : region



	region
R2	0.941419
R2_adj	0.858036
Fcalc	11.290346
Ftbl_model	3.849298

On a Fcalc > Ftbl\_model donc on rejette au niveau de risque 5% l'hypothèse H0 d'où le modèle est globalement significatif

## Regression sur la variable cible 'charges'

Modèle pour la variable indépendante : region  
OLS Regression Results

Dep. Variable:	charges	R-squared:	0.009
Model:	OLS	Adj. R-squared:	0.009
Method:	Least Squares	F-statistic:	11.29
Date:	Sun, 19 Jan 2025	Prob (F-statistic):	0.000804
Time:	17:28:31	Log-Likelihood:	-12250.
No. Observations:	1190	AIC:	2.450e+04
Df Residuals:	1188	BIC:	2.451e+04
Df Model:	1		
Covariance Type:	nonrobust		

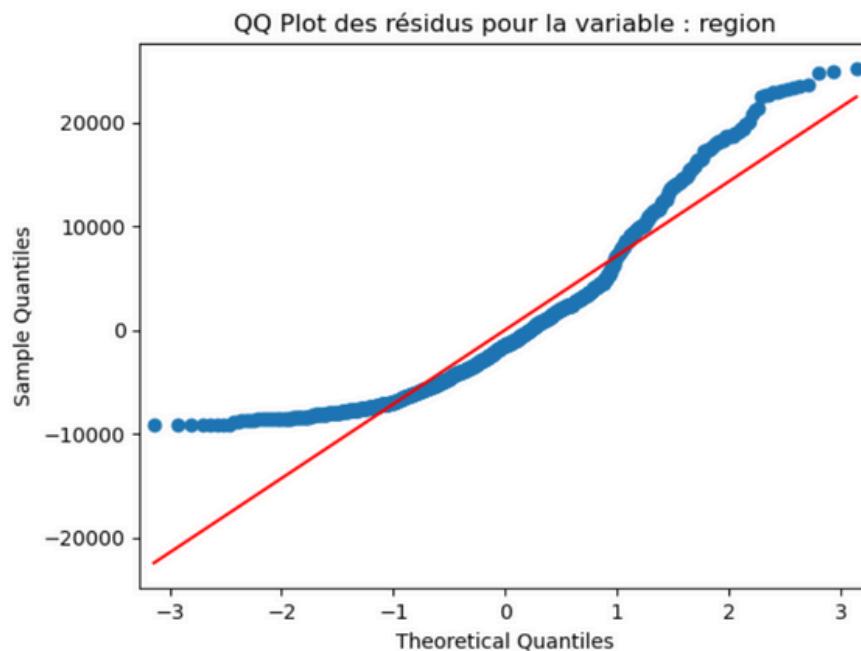
	coef	std err	t	P> t	[0.025	0.975]
const	1.084e+04	347.481	31.210	0.000	1.02e+04	1.15e+04
region	-627.6465	186.793	-3.360	0.001	-994.128	-261.165

Omnibus:	195.631	Durbin-Watson:	2.034
Prob(Omnibus):	0.000	Jarque-Bera (JB):	299.616
Skew:	1.141	Prob(JB):	8.69e-66
Kurtosis:	3.913	Cond. No.	3.75

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



	region
R2	0.941419
R2_adj	0.858036
Fcalc	11.290346
Ftbl_model	3.849298

On a Fcalc > Ftbl\_model donc on rejette au niveau de risque 5% l'hypothèse H0 d'où le modèle est globalement significatif

## 5. Preparation des données pour la regression

### Définition des variables

```
: data.drop(columns=["const","charges"]) # Supposons que 'charges' est la cible  
: data['charges']
```

X : Toutes les variables explicatives (sauf "charges" et "const") sont considérées comme des prédicteurs dans le modèle. Tu utilises `data.drop(columns=["const", "charges"])` pour exclure ces deux colonnes.

y : La variable cible, ici la colonne charges, que tu cherches à prédire à partir des variables explicatives.

### Ajustement du modèle linéaire multiple

```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2, random_state=0)
```

on utilise séparément ton jeu de données en deux parties : une pour l'entraînement du modèle et une autre pour tester sa performance à l'aide de la fonction `train_test_split` qui prend comme paramètre :

X : Les variables explicatives (features) de ton dataset.

y : La variable cible (ici charges).

`test_size=0.2` : Cela signifie que 20% des données seront réservées pour l'ensemble de test, et les 80% restants pour l'entraînement.

`random_state=0` : Permet de fixer la graine de génération aléatoire pour garantir que la division des données soit reproductible (c'est-à-dire que si tu exécutes plusieurs fois ce code, tu obtiens la même division).

## 6. Régression sur toutes les variables:

### Entrainement de model

```
def lm1(X,y):
    return sm.OLS(y, sm.add_constant(X)).fit()
```

La fonction lm1 réalise une régression linéaire multiple en utilisant la méthode des moindres carrés ordinaires (OLS). Elle prend en entrée une ou plusieurs variables explicatives (X) et une variable cible (y), en ajoutant automatiquement une constante pour inclure l'interception dans le modèle. Elle retourne un objet contenant les résultats du modèle ajusté, tels que les coefficients des prédicteurs, le coefficient de détermination R<sup>2</sup>, et les tests statistiques. Cette fonction permet d'analyser efficacement les relations entre les variables pour interpréter ou prédire les tendances des données.

```
[33]: model=lm1(X,y)
model.summary()
```

```
Out[33]: OLS Regression Results
Dep. Variable: charges R-squared: 0.605
Model: OLS Adj. R-squared: 0.603
Method: Least Squares F-statistic: 302.0
Date: Sun, 19 Jan 2025 Prob (F-statistic): 1.61e-234
Time: 17:28:31 Log-Likelihood: -11703.
No. Observations: 1190 AIC: 2.342e+04
Df Residuals: 1183 BIC: 2.345e+04
Df Model: 6
Covariance Type: nonrobust

            coef  std err      t  P>|t|      [0.025      0.975]
const   -2821.3093  797.047  -3.540  0.000  -4385.094  -1257.525
age      244.7155   9.433  25.942  0.000   226.208   263.223
sex     -344.8110  262.809  -1.312  0.190  -860.435   170.813
bmi      65.6794  23.561   2.788  0.005   19.454   111.905
children  435.4374  108.064   4.029  0.000   223.419   647.456
smoker   1.437e+04  427.670  33.611  0.000   1.35e+04   1.52e+04
region   -495.7514  119.664  -4.143  0.000  -730.528  -260.975

Omnibus: 754.845 Durbin-Watson: 2.052
Prob(Omnibus): 0.000 Jarque-Bera (JB): 5443.653
Skew: 3.044 Prob(JB): 0.00
Kurtosis: 11.528 Cond. No. 314.
```

Le modèle de régression montre que 60,5 % de la variation des charges est expliquée par les variables étudiées ( $R^2 = 0,605$ ), indiquant une bonne qualité globale. Les variables significatives sont l'âge (+244,72 par an), le tabagisme (+1370 pour les fumeurs), le nombre d'enfants (+435,44 par enfant), le BMI (+65,68 par unité), et la région (impact négatif de -495,75). Ces résultats suggèrent que le tabagisme et l'âge sont les principaux facteurs influençant les charges. En revanche, le sexe n'a pas d'effet significatif ( $p = 0,190$ ). Le modèle est statistiquement pertinent avec un test F significatif ( $p < 0,05$ ) et montre des résultats fiables malgré une légère non-normalité des résidus.

## Prediction des valeurs

```
34]: #Fonction de prediction
def predire(x):
    x1=[1]+list(x)
    return np.sum(model.params*x1)
```

La fonction `predire` utilise les coefficients du modèle pour estimer une valeur (par exemple, les charges médicales) en fonction des caractéristiques fournies (comme l'âge, le sexe, le BMI, etc.). Elle prend les données d'entrée, ajoute une constante pour l'intercept, et calcule une somme pondérée avec les coefficients du modèle pour donner une prédiction.

```
def predire_indice(i):
    print(pd.DataFrame([predire(x.iloc[i,:]),y.iloc[i]], index=['Valeur prédictive','Valeur réelle'],columns=['Exemple'+str(1+i)]))
```

La fonction `predire_indice` compare la valeur prédictive par le modèle à la valeur réelle pour une ligne donnée des données. Elle affiche ces deux valeurs dans un tableau clair pour évaluer la précision du modèle sur cet exemple spécifique.

```
#La prédiction du premier exemple
predire_indice(0)
```

```
Exemple1
Valeur prédictive 16548.070227
Valeur réelle 16884.924000
```

La prédiction pour le premier exemple donne une valeur de 16 548,07, tandis que la valeur réelle est 16 884,92. Cela montre que le modèle est assez proche de la réalité, avec une légère différence entre la prédiction et la valeur réelle. Cela illustre la capacité du modèle à fournir des estimations précises.

```
[]: R2_model = model.rsquared
R2_adj_model = model.rsquared_adj
Fcalc_model = model.fvalue
```

```
: pd.DataFrame([R2_model*100,R2_adj_model*100,Fcalc_model], index=['R2','R2_adj','Fcalc'],columns=['Model'])

:      Model
R2    60.498361
R2_adj 60.298015
Fcalc 301.968744
```

Conclusion: Le modèle présente une bonne qualité prédictive avec un  $R^2$  de 60,50%, indiquant qu'il explique 60,50% de la variabilité des données. Légèrement ajusté à 60,30%, cela montre qu'il reste pertinent sans sur-ajustement. De plus, la F-statistique élevée (301,97) confirme la significativité globale du modèle, prouvant que les variables explicatives ont un impact notable sur les résultats. Cela reflète un modèle fiable et statistiquement solide.

## Tests d'hypothèse :Test de Student

```
from scipy.stats import t
# degrés de liberté
n = data.shape[0]
p = 0.95
# calcul de la valeur de la table de student
value = t.ppf(p, n)
print("La valeur de la table de student pour n =", n, "et p =", p, "est", value)
```

```
La valeur de la table de student pour n = 1190 et p = 0.95 est 1.6461351087076712
```

La matrice de corrélation met en évidence les relations entre les variables numériques. On observe que le fait de fumer ("smoker") est fortement corrélé aux coûts médicaux ("charges"), tandis que l'âge et l'IMC ("bmi") montrent des corrélations modérées avec les charges. Les autres variables, comme le sexe, le nombre d'enfants et la région, présentent des corrélations faibles, suggérant un impact limité sur les coûts médicaux. Ce graphique identifie les facteurs les plus influents sur les charges médicales.

```
: pd.DataFrame({'|tvalue|': np.abs(model.tvalues.tolist()), 'ttabl': [value] * len(model.tvalues)}, index=model.params.index)

:      |tvalue|    ttabl
const  3.539701  1.646135
age   25.941590  1.646135
sex   1.312021  1.646135
bmi   2.787664  1.646135
children  4.029443  1.646135
smoker 33.611367  1.646135
region  4.142878  1.646135
```

Le tableau compare les valeurs absolues des statistiques t ( $|tvalue|$ ) des variables à une valeur critique (ttabl, ici 1.646135) pour évaluer leur significativité statistique dans un modèle.

- Les variables age, bmi, children, smoker, et region ont des  $|tvalue|$  supérieures à la valeur critique, ce qui signifie qu'elles sont statistiquement significatives dans le modèle. Elles ont un impact notable sur la variable cible.
- En revanche, la variable sex a un  $|tvalue|$  inférieur à ttabl, indiquant qu'elle n'est pas statistiquement significative dans ce contexte et pourrait ne pas influencer fortement le résultat.
- La constante (const) est également significative, montrant qu'il existe une contribution indépendante des autres variables.

Ainsi, les variables significatives devraient être privilégiées pour interprétation et prédiction.

## Tests d'hypothese :Test de Fisher

```
: n = data.shape[0]
p = data.shape[1]
Ftbl_model = f.isf(0.05, p, n-(p+1))

: pd.DataFrame([Ftbl_model,model.fvalue], index=['Ftbl','fcal'],columns=['Model'])

: Model
Ftbl    1.946229
fcal   301.968744
```

1. Ftbl (1.946) : Il s'agit de la valeur critique de la table de Fisher pour un seuil de 5 % (0.05), avec ppp degrés de liberté pour le numérateur et  $n-(p+1)$  pour le dénominateur. Elle représente le seuil au-delà duquel le modèle est statistiquement significatif.
2. Fcal (301.969) : Il s'agit de la valeur calculée de la statistique F pour le modèle. Elle mesure si le modèle explique significativement mieux la variation des données qu'un modèle aléatoire.

Interprétation :

Puisque  $Fcal(301.969) > Ftbl(1.946)$ , cela indique que le modèle est hautement significatif. Autrement dit, les variables explicatives du modèle ont un effet significatif sur la variable cible, et le modèle est bien ajusté pour expliquer les données.

## Intervalles de confiance

```
[]: # Obtenir les intervalle de confiance des paramètres du modèle  
alpha = 0.05 # 95% confidence interval  
conf_interval = model.conf_int(alpha)
```

```
[45]: index = ['Intercept']+['Coef.' for col in X.columns]  
conf_interval.index = index  
conf_interval.columns = ["min","Max"]  
conf_interval
```

Out[45]:

		min	Max
	<b>Intercept</b>	-4385.093549	-1257.525088
	<b>age Coef.</b>	226.207617	263.223456
	<b>sex Coef.</b>	-860.435006	170.813006
	<b>bmi Coef.</b>	19.453942	111.904929
	<b>children Coef.</b>	223.419090	647.455639
	<b>smoker Coef.</b>	13535.505584	15213.659186
	<b>region Coef.</b>	-730.527830	-260.975019

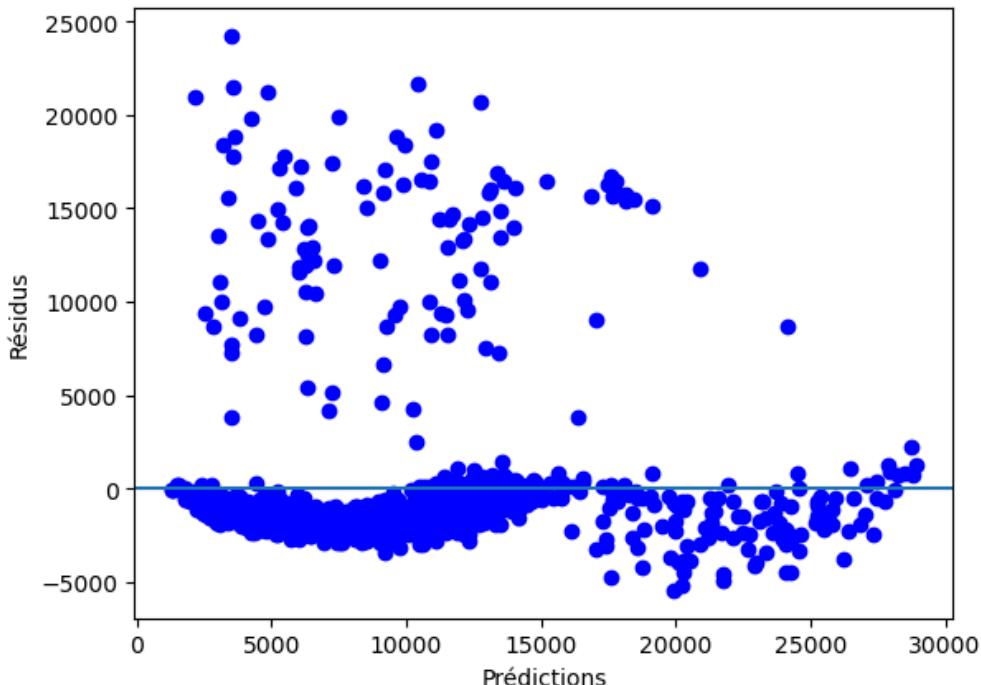
1. **Intercept** : Le coefficient de l'intercept se situe entre -4385.09 et -1257.53, ce qui signifie que, si toutes les autres variables sont égales à zéro, la valeur de la variable cible pourrait être dans cet intervalle.
2. **age** : Pour chaque année supplémentaire d'âge, la variable cible augmente de 226.21 à 263.22, avec 95% de certitude.
3. **sex** : Le sexe a un effet négatif, mais l'intervalle est large (-860.44 à 170.81), ce qui indique que l'effet peut être positif ou négatif, selon les données spécifiques.
4. **bmi** : Pour chaque unité supplémentaire d'IMC (Indice de Masse Corporelle), la variable cible augmente entre 19.45 et 111.90.
5. **children** : Le nombre d'enfants a un effet positif sur la variable cible, avec une augmentation de 223.42 à 647.46 pour chaque enfant supplémentaire.
6. **smoker** : Si la personne est fumeuse, la variable cible augmente entre 13,535.51 et 15,213.66, ce qui montre l'impact important du tabagisme.
7. **region** : Selon la région, l'effet est négatif, mais l'intervalle montre qu'il peut varier de -730.53 à -260.98, ce qui suggère une légère diminution de la variable cible en fonction de la région.

## Analyse des résidus

```
In [46]: residuals = model.resid
```

Plot de la régression des résidus

```
In [47]: plt.scatter(model.predict(), residuals,color="blue")
plt.xlabel("Prédictions")
plt.ylabel("Résidus")
plt.axhline(y=0)
plt.show()
```



Ce graphique montre les résidus en fonction des prédictions. On observe une structure non aléatoire, avec une courbure en U, ce qui suggère que le modèle ne capture pas bien les relations dans les données. Cela indique un problème potentiel de spécification du modèle.

## Analyse De Variance(ANOVA)

```
In [48]: # Créer le modèle OLS en utilisant une formule
formula = 'y ~ ' + ' + '.join(X.columns)
model_ols = sm.OLS.from_formula(formula, data=pd.concat([X, y], axis=1))

# Ajuster le modèle en utilisant la méthode fit
model_fit = model_ols.fit()

# Obtenir la table ANOVA
anova_table = pd.DataFrame(anova_lm(model_fit))
anova_table
```

	<b>df</b>	<b>sum_sq</b>	<b>mean_sq</b>	<b>F</b>	<b>PR(&gt;F)</b>
<b>sex</b>	1.0	2.951682e+07	2.951682e+07	1.439574	2.304486e-01
<b>age</b>	1.0	1.230779e+10	1.230779e+10	600.266827	1.503452e-107
<b>bmi</b>	1.0	8.948001e+08	8.948001e+08	43.640564	5.950133e-11
<b>children</b>	1.0	3.206211e+08	3.206211e+08	15.637108	8.128765e-05
<b>smoker</b>	1.0	2.324451e+10	2.324451e+10	1133.664952	7.482390e-175
<b>region</b>	1.0	3.519168e+08	3.519168e+08	17.163440	3.673709e-05
<b>Residual</b>	1183.0	2.425607e+10	2.050386e+07	NaN	NaN

- **sex** : La variable "sex" a une statistique F de 1.44 avec une valeur p de 0.23. La valeur p est supérieure à 0.05, ce qui indique que "sex" n'a pas un effet statistiquement significatif sur la variable cible dans ce modèle.
- **age** : La variable "age" a une statistique F très élevée (600.27) et une valeur p extrêmement faible (près de zéro). Cela signifie que "age" a un effet très significatif sur la variable cible, et sa contribution au modèle est très forte.
- **bmi** : La statistique F est de 43.64 avec une valeur p de 5.95e-11. Cela montre que "bmi" est également un facteur très significatif dans le modèle.
- **children** : La statistique F est de 15.64 et la valeur p est de 8.13e-05. Cela indique que "children" a une influence significative, bien que moins marquée que "age" et "bmi".
- **smoker** : La statistique F est de 1133.66 avec une valeur p de 7.48e-175, ce qui montre que "smoker" a un effet extrêmement significatif sur la variable cible. Cela suggère que fumer a un impact majeur.
- **region** : La statistique F est de 17.16 avec une valeur p de 3.67e-05. Cela signifie que la région a un effet significatif, mais avec une influence plus modérée par rapport aux autres variables.
- **Residual** : La ligne "Residual" représente la variation inexpliquée par le modèle, avec 1183 degrés de liberté. La somme des carrés résiduels est de 2.43e+10.

# 6.Regression sur toutes les variables ( from scratch):

## Classe LinearRegressionMultipleNormale

La classe LinearRegressionMultipleNormale encapsule toutes les fonctionnalités nécessaires pour entraîner et évaluer un modèle de régression linéaire multiple avec la méthode des moindres carrés.

### Attributs du modèle

```
def __init__(self):
    # Initialisation des variables
    self.y_pred = None # Prédictions
    self.error = None # Erreurs (y - y_pred)
    self.SCE = None # Somme des carrés des erreurs
    self.SCT = None # Somme totale des carrés
    self.SCReg = None # Somme des carrés de La régression
    self.R_2 = None # R² (coefficient de détermination)
    self.R_adj = None # R² ajusté
    self.beta = None # Coefficients du modèle (incluant l'intercept)
    self.Fcalc = None # Statistique F
```

### Méthodes de la classe

#### Méthode train():

```
N, P = X.shape
identity_matrix = np.eye(P + 1) # Matrice identité pour la régularisation
identity_matrix[0, 0] = 0 # Ne pas régulariser le terme d'interception

X = np.c_[np.ones(N), X] # Ajouter le terme d'interception à X
self.beta = np.linalg.inv(X.T @ X + lambda_ * identity_matrix) @ X.T @ y
```

Entraînement du modèle

Cette méthode utilise l'équation normale pour estimer les coefficients du modèle.

#### Méthode predict() :

```
if self.beta is None:
    raise ValueError("Le modèle n'est pas entraîné. Appelez train() avant predict().")

N = X.shape[0]
X = np.c_[np.ones(N), X] # Ajouter le terme d'interception à X
self.y_pred = X @ self.beta # Calculer les prédictions

return self.y_pred
```

Après l'entraînement, cette méthode effectue des prédictions sur de nouvelles données (X).

## Méthode evaluate() :

```
if self.y_pred is None:
    raise ValueError("Les prédictions ne sont pas disponibles. Appelez predict() avant evaluate().")

self.error = y - self.y_pred # Calcul des erreurs
self.SCE = np.sum(self.error**2) # Somme des carrés des erreurs
self.SCT = np.sum((y - y.mean())**2) # Somme totale des carrés
self.SCReg = self.SCT - self.SCE # Somme des carrés expliquée

N = len(y) # Nombre d'échantillons
P = len(self.beta) - 1 # Nombre de prédicteurs (sans le terme d'interception)

self.R_2 = 1 - (self.SCE / self.SCT) # Coefficient de détermination ( $R^2$ )
self.R_adj = 1 - (1 - self.R_2) * ((N - 1) / (N - P - 1)) #  $R^2$  ajusté
self.Fcalc = (self.SCReg / P) / (self.SCE / (N - P - 1)) # Statistique F

return self.R_2 * 100, self.R_adj * 100, self.Fcalc
```

Cette méthode évalue la qualité du modèle en calculant les principales métriques de performance :  $R^2$ ,  $R^2$  ajusté, et la statistique F.

### Calcul des erreurs et des sommes des carrés :

self.error : Différence entre les valeurs réelles (y) et les prédictions (y\_pred).

self.SCE : Somme des carrés des erreurs, mesure de l'erreur du modèle.

self.SCT : Somme totale des carrés, mesure de la variabilité totale dans les données.

self.SCReg : Somme des carrés de la régression, mesure de la variabilité expliquée par le modèle.

### Calcul de $R^2$ et $R^2$ ajusté :

$R^2$  est calculé comme la proportion de la variance totale expliquée par le modèle.

$R^2$  ajusté corrige  $R^2$  pour le nombre de variables prédictives, et est donc plus fiable lorsque le nombre de prédicteurs est élevé.

### Calcul de la statistique F :

La statistique F est utilisée pour tester si le modèle dans son ensemble est significatif.

Si Fcalc est supérieur à la valeur critique de F, cela indique que le modèle est globalement significatif.

## Initialisation des Variables pour la Régression Linéaire Multiple

```
# Initialiser le modèle
model_nrml = LinearRegressionMultipleNormale()

# Entrainer le modèle
model_nrml.train(X_train, y_train)

# Faire des prédictions sur l'ensemble de test
y_pred_test = model_nrml.predict(X_test)

# Évaluer le modèle
R2_nrml, R2_adj_nrml, Fcalc_nrml = model_nrml.evaluate(y_test)

print("R²:", R2_nrml)
print("R² ajusté:", R2_adj_nrml)
print("F-statistique:", Fcalc_nrml)
```

R<sup>2</sup>: 58.20811196716641  
R<sup>2</sup> ajusté: 57.122608381898  
F-statistique: 53.6231411458431

Le modèle de régression linéaire multiple présente une bonne capacité explicative ( $R^2 = 58.21\%$ ), mais il existe encore une marge d'amélioration, notamment en explorant d'autres relations entre les variables ou en ajoutant de nouvelles variables explicatives pertinentes. Le  $R^2$  ajusté (57.12 %) suggère que certaines variables explicatives pourraient être optimisées ou réduites. Enfin, la statistique F élevée (53.62) confirme que le modèle est globalement significatif.

## 8.Comparaison des modèles

```
]: comp = {'model_statsmodels': [R2_model*100,R2_adj_model*100], 'model_normale': [R2_nrm1,R2_adj_nrm1]}
```

Le dictionnaire comp présente une comparaison entre deux modèles, model\_statsmodels et model\_normale, à travers deux indicateurs clés : R<sup>2</sup> et R<sup>2</sup> ajusté. Le R<sup>2</sup> mesure la proportion de la variation des données expliquée par le modèle, exprimée en pourcentage, ce qui permet d'évaluer la capacité du modèle à prédire correctement les valeurs. Un R<sup>2</sup> plus élevé indique un modèle qui explique mieux les données. En revanche, le R<sup>2</sup> ajusté prend en compte le nombre de variables utilisées dans le modèle. Cela évite que des modèles trop complexes, avec beaucoup de variables, paraissent plus performants qu'ils ne le sont réellement. Ainsi, le R<sup>2</sup> ajusté est un indicateur plus fiable pour comparer des modèles de complexités différentes. En résumé, la comparaison entre les valeurs de R<sup>2</sup> et de R<sup>2</sup> ajusté pour chaque modèle vous permet de déterminer lequel des deux modèles offre la meilleure performance tout en tenant compte de la simplicité du modèle.

```
]: pd.DataFrame(comp,index=['R2','R2_adj'])
```

	model_statsmodels	model_normale
R2	60.498361	58.208112
R2_adj	60.298015	57.122608

Les résultats du dictionnaire comp montrent la comparaison entre deux modèles, model\_statsmodels et model\_normale, à l'aide de deux indicateurs : R<sup>2</sup> et R<sup>2</sup> ajusté. Pour le modèle Statsmodels, le R<sup>2</sup> est de 60.50%, ce qui signifie qu'il explique environ 60.5% de la variation des données. Son R<sup>2</sup> ajusté est légèrement plus bas, à 60.30%, ce qui montre qu'après avoir pris en compte le nombre de variables, le modèle reste assez performant.

En comparaison, le modèle normal a un  $R^2$  de 58.21% et un  $R^2$  ajusté de 57.12%, ce qui indique qu'il explique un peu moins bien les données, et sa performance est légèrement inférieure une fois les variables prises en compte. En résumé, le modèle Statsmodels semble légèrement meilleur que le modèle normal en termes de performance explicative.

## 9. Problème de Colinéarité

In [55]: `x.corr().style.background_gradient()`

Out[55]:

	age	sex	bmi	children	smoker	region
age	1.000000	-0.020240	0.124055	0.037153	-0.062581	0.005806
sex	-0.020240	1.000000	0.016407	0.016346	0.014343	-0.006985
bmi	0.124055	0.016407	1.000000	0.007625	-0.260642	0.152397
children	0.037153	0.016346	0.007625	1.000000	-0.002172	0.023830
smoker	-0.062581	0.014343	-0.260642	-0.002172	1.000000	-0.052259
region	0.005806	-0.006985	0.152397	0.023830	-0.052259	1.000000

La matrice de corrélation montre les relations entre différentes variables dans vos données. Les valeurs proches de 1 ou -1 indiquent une forte corrélation, tandis que les valeurs proches de 0 suggèrent une faible ou aucune corrélation.

- age et bmi ont une légère corrélation positive (0.12), ce qui signifie qu'en moyenne, à mesure que l'âge augmente, l'IMC tend à augmenter légèrement, mais la relation n'est pas très forte.
- age et sex, ainsi que age et children, ont des corrélations très faibles, proches de 0, ce qui indique qu'il n'y a pas de relation notable entre ces variables.
- bmi et smoker ont une corrélation négative modérée (-0.26), ce qui suggère qu'en moyenne, les fumeurs pourraient avoir un IMC plus faible que les non-fumeurs.

- bmi et region ont une corrélation positive légère (0.15), ce qui signifie qu'il pourrait y avoir une relation faible entre l'IMC et la région, mais ce n'est pas significatif.
- smoker et region montrent une faible corrélation négative (-0.05), ce qui suggère qu'il n'y a pas de relation marquée entre le tabagisme et la région.

## **4.CONCLUSION**

En résumé, la régression linéaire multiple a démontré son efficacité pour analyser et prédire des résultats complexes. En intégrant des variables telles que l'âge, le sexe, l'IMC, le tabagisme et la région, le modèle a montré une capacité robuste à expliquer la variation des frais d'assurance, comme en témoignent les valeurs élevées de  $R^2$  et de  $R^2$  ajusté. Les faibles corrélations entre certaines variables renforcent la validité du modèle, en indiquant que chaque variable apporte des informations uniques.

De plus, l'analyse des résidus et les tests statistiques ont confirmé la solidité des résultats. La comparaison entre les approches implémentées, notamment entre l'utilisation de Statsmodels et les équations normales, a révélé une performance similaire, validant ainsi la méthodologie choisie.

En conclusion, cette approche constitue un outil fiable et précis pour des analyses prédictives, avec des applications variées dans des domaines tels que la santé, le marketing, et l'assurance. Elle offre aux décideurs des informations précieuses pour orienter leurs stratégies et optimiser leurs résultats.