Rapport Projet Python – Ingéniérie des données

Table des matières

[Analyse de survie 1](#_Toc134022441)

[Présentation du langage utilisé : 1](#_Toc134022442)

[Présentation des bibliothèques utilisées : 1](#_Toc134022443)

[Pandas : 1](#_Toc134022444)

[LifeLines : 1](#_Toc134022445)

[Streamlit : 2](#_Toc134022446)

[Plotly : 2](#_Toc134022447)

[Matplotlib : 3](#_Toc134022448)

[Structure de l’application : 3](#_Toc134022449)

[Descriptif de l’application : 3](#_Toc134022450)

[Décrire les possibilités d’hébergement web : 3](#_Toc134022451)

[Ce qui ne fonctionne pas : 3](#_Toc134022452)

[Difficultés rencontrées : 3](#_Toc134022453)

# Analyse de survie

Ce document présente le travail effectué par Maxime Dangelser et Alexis Aulagnier dans le cadre du cours de Projet d’ingénierie des données du Master 1 MIAGE de Polytech Lyon.

Le sujet porte sur l’analyse de survie avec la bibliothèque lifeline en python.

# Présentation du langage utilisé :

Lors de ce projet nous avons utilisé le langage de programmation Python.

Python est un langage de programmation interprété, à syntaxe claire et simple, et qui met l'accent sur la lisibilité du code. Il est facile à apprendre et à utiliser pour les débutants en programmation, mais aussi assez puissant pour être utilisé dans des projets complexes.

Python est un langage polyvalent, utilisé dans une grande variété d'applications, notamment pour la science des données, l'apprentissage automatique, la création d'applications web, l'automatisation de tâches, les scripts système, les jeux et bien plus encore. Il a une grande communauté de développeurs qui ont créé une multitude de bibliothèques et de frameworks, tels que Pandas, Numpy, Matplotlib, Django, Flask, etc., qui facilitent le développement de projets de différentes tailles et complexités.

En outre, Python est un langage open-source, ce qui signifie que tout le monde peut l'utiliser, le modifier et le distribuer librement. C'est un langage très populaire et largement utilisé dans l'industrie, l'éducation et la recherche.

# Présentation des bibliothèques utilisées :

Pour effectuer ce travail nous avons utilisé différentes bibliothèques présentées ci-dessous.

## Pandas :

Pandas est une bibliothèque open-source pour Python qui permet de manipuler et d'analyser des données tabulaires. Pandas fournit des fonctionnalités pour lire et écrire des données dans différents formats de fichiers, tels que CSV, Excel, JSON, SQL et bien plus encore. Elle permet également de nettoyer et de préparer les données pour l'analyse, notamment en gérant les valeurs manquantes et en effectuant des opérations de fusion, de regroupement et de filtrage.

## LifeLines :

Lifelines est une bibliothèque open-source de Python qui fournit des outils pour l'analyse de données de survie et l'estimation de la durée de vie. Elle est conçue pour les scientifiques des données, les ingénieurs et les chercheurs qui travaillent avec des données de survie dans divers domaines tels que la médecine, l'économie, l'ingénierie, la biologie et bien d'autres encore. Lifelines permet de modéliser et d'analyser des données de survie à l'aide de techniques telles que l'estimation de la fonction de survie, l'analyse de la durée de vie, les courbes de Kaplan-Meier, les modèles de régression Cox et bien d'autres encore. Elle fournit également des fonctionnalités pour l'analyse des données censurées, telles que les données de survie tronquées, les données de survie avec des valeurs manquantes, les données de survie intervalles et bien d'autres encore.

Nous utilisons 2 classes :

* **KaplanMeierFitter** est une classe de Lifelines qui permet de constuire une courbe de survie en tenant compte des données censurées, c'est-à-dire les individus qui quittent l'étude sans avoir connu l'événement d'intérêt (ici la mort).
* **CoxPHFitter** est une autre classe de Lifelines qui implémente la régression de Cox. Elle permet de modéliser une relation entre les variables explicatives et le risque relatif de décès.

## Streamlit :

Streamlit est une bibliothèque open-source de Python qui permet de créer facilement des applications web interactives à partir de scripts Python. Elle fournit une interface simple et intuitive pour créer des applications web en utilisant des commandes simples telles que "st.title", "st.write", "st.plot" et bien d'autres encore. Streamlit permet aux utilisateurs de créer des applications web interactives à partir de scripts Python en quelques minutes, sans avoir à apprendre des langages de programmation web tels que HTML, CSS ou JavaScript.

Elle prend également en charge les graphiques interactifs, les cartes, les animations, les widgets, les formulaires et bien d'autres encore. Streamlit est très populaire dans la communauté des scientifiques des données, car elle permet de créer des tableaux de bord interactifs pour visualiser et explorer des données rapidement et facilement. Elle est également très utile pour la création de prototypes, la démonstration de concepts, la formation et l'enseignement.

## Plotly :

**Plotly** est une bibliothèque de visualisation de données en Python, utilisée pour créer des graphiques interactifs. Elle propose deux interfaces pour créer des graphiques : plotly.express et plotly.graph\_objects.

**Plotly**.**express** est une interface haut niveau pour créer des graphiques rapidement et facilement. Elle est construite sur la base de la bibliothèque pandas et est conçue pour être facile à utiliser pour les débutants en visualisation de données. Elle fournit une grande variété de graphiques prêts à l'emploi pour les types de données les plus courants, tels que les graphiques en barres, les graphiques à secteurs, les graphiques en nuage de points, les diagrammes en boîte, les histogrammes, etc. Il est également possible de personnaliser ces graphiques en ajoutant des options pour la couleur, la taille, les axes, etc.

**Plotly.graph\_objects** est une interface bas niveau pour créer des graphiques plus personnalisés. Elle est conçue pour les utilisateurs avancés qui souhaitent un contrôle total sur leur graphique. Cette interface permet de créer des graphiques à partir de zéro en ajoutant des éléments tels que des axes, des titres, des annotations, des légendes, des formes et des images. Elle permet également de créer des graphiques interactifs en ajoutant des événements, des animations et des liens.

## Matplotlib :

Matplotlib est une bibliothèque de visualisation de données en Python qui permet de créer des graphiques statiques. Matplotlib.pyplot, souvent abrégé en plt, est un module de Matplotlib qui fournit une interface de programmation similaire à celle de MATLAB pour créer des graphiques en utilisant des commandes simples.

# Structure de l’application :

## Organisation des fichiers :

Il y a 4 fichiers python :

* main.py : Ce fichier contient l’ensemble du code de l’application et c’est lui que l’on voit lorsqu’on démarre l’application.
* constant.py : Ce fichier contient les constantes que nous utilisons dans le code. Pour faciliter la modification en cas de besoin.
* text.py : Ce fichier contient le texte affiché dans l’application pour alléger le fichier main.py.
* code\_text.py : Ce fichier contient le texte du code affichés dans l’application via l’option « Montrer le code ». Également pour alléger le fichier main.py.

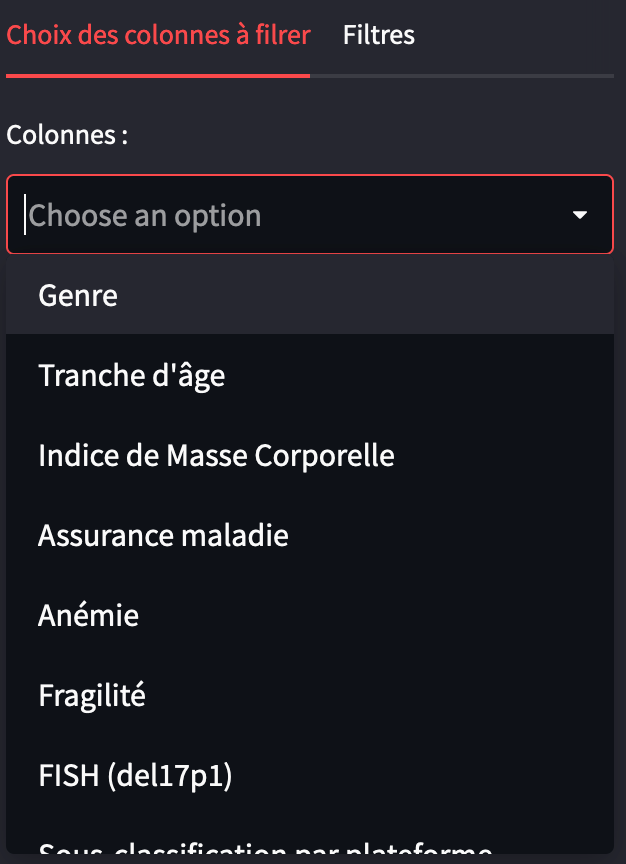
Le dossier « données » qui contient les 2 fichiers de données :

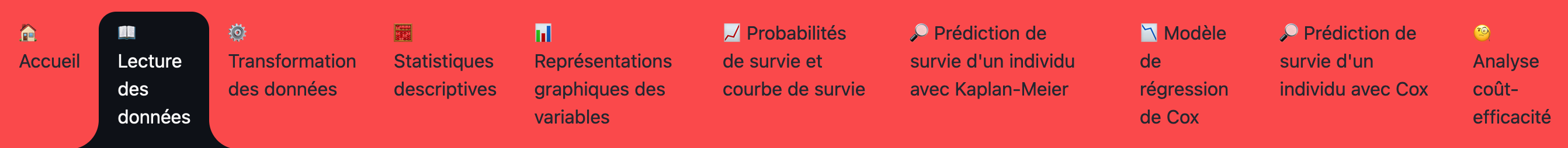
* MockPatientDatabaseOscar.csv : Le fichier de données d’origine, sans modifications.
* MockPatientDatabaseOscar – Modified.csv : Le fichier de données avec 300 lignes de données que nous avons généré en dehors du programme.

Le fichier requirements.txt qui permet l’installation des librairies utilisées.

## Structure de l’application :

Le fichier main comprend plusieurs fonctions :

* Gestion du menu latéral :
  + left\_menu : Propose un choix de colonnes à filtrer en 2 onglets et renvoie les filtres au format dictionnaire :  
    
  + left\_menu\_choice\_for\_regression\_model : Propose un choix de 4 colonnes maximum pour le modèle de régression de Cox (Le menu est affiché uniquement lors de l’utilisation du modèle). Une image contenant texte

    Description générée automatiquement
* Gestion du menu supérieur :
  + top\_menu : Cette fonction est la fonction principale de l’application puisque c’est elle qui gère le menu supérieur, et les pages. Elle appelle les fonctions utilisées pour chaque page.  
    
* Autres :
  + Toutes les autres fonctions sont utilisées pour calculer et afficher les éléments pour chaque page de l’application.

# Descriptif de l’application :

## Accueil :

Affiche le texte de présentation des librairies et du sujet.

## Lecture des données :

Présente les données d’origine sous forme de table et avec une explication pour les colonnes principales.

## Transformation des données :

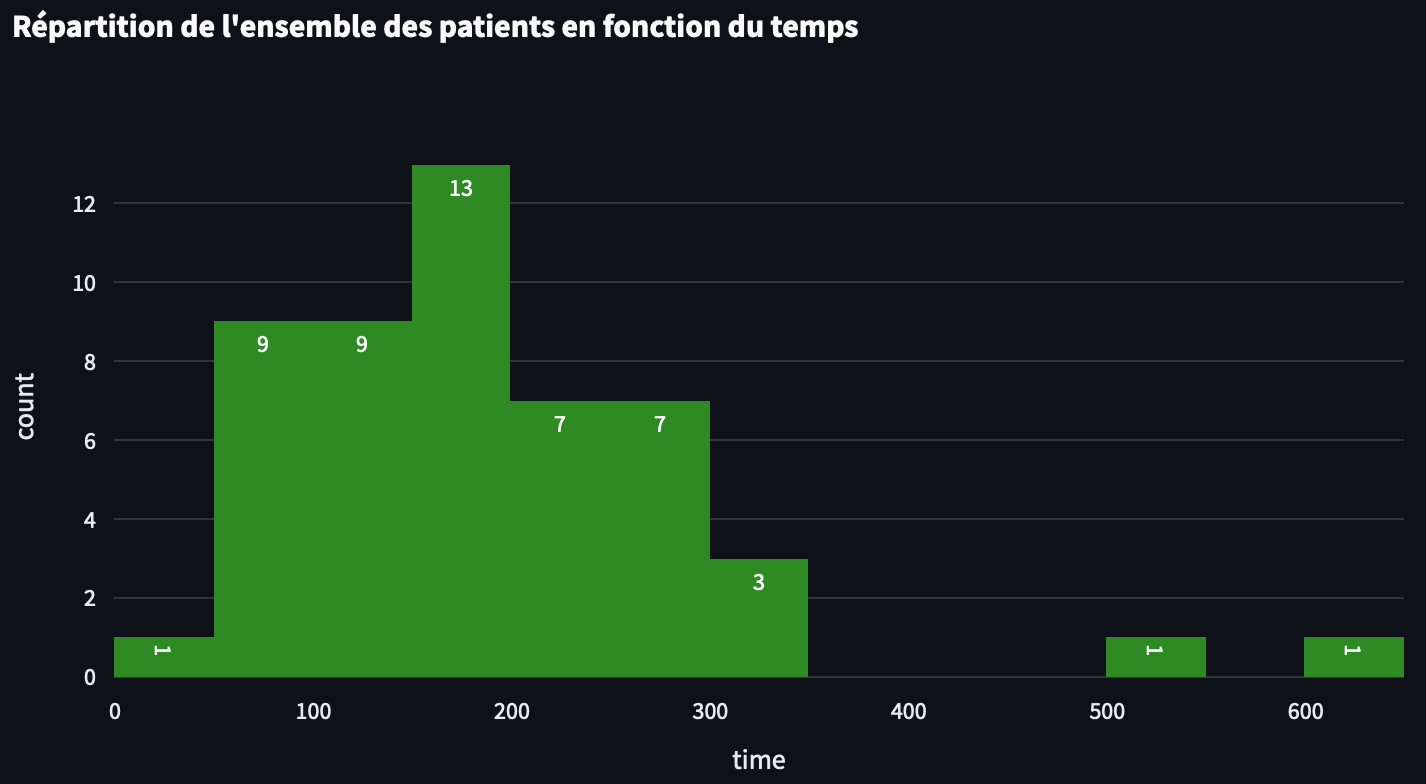
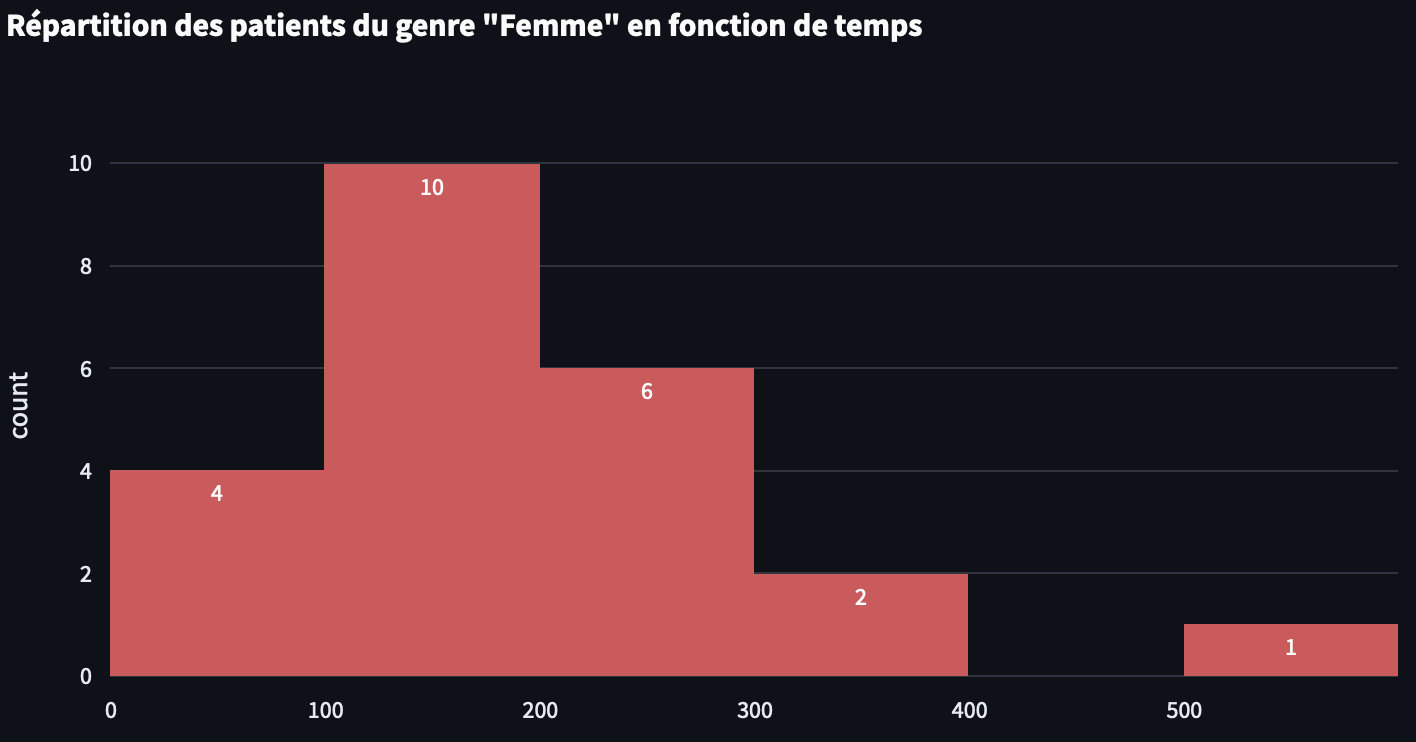
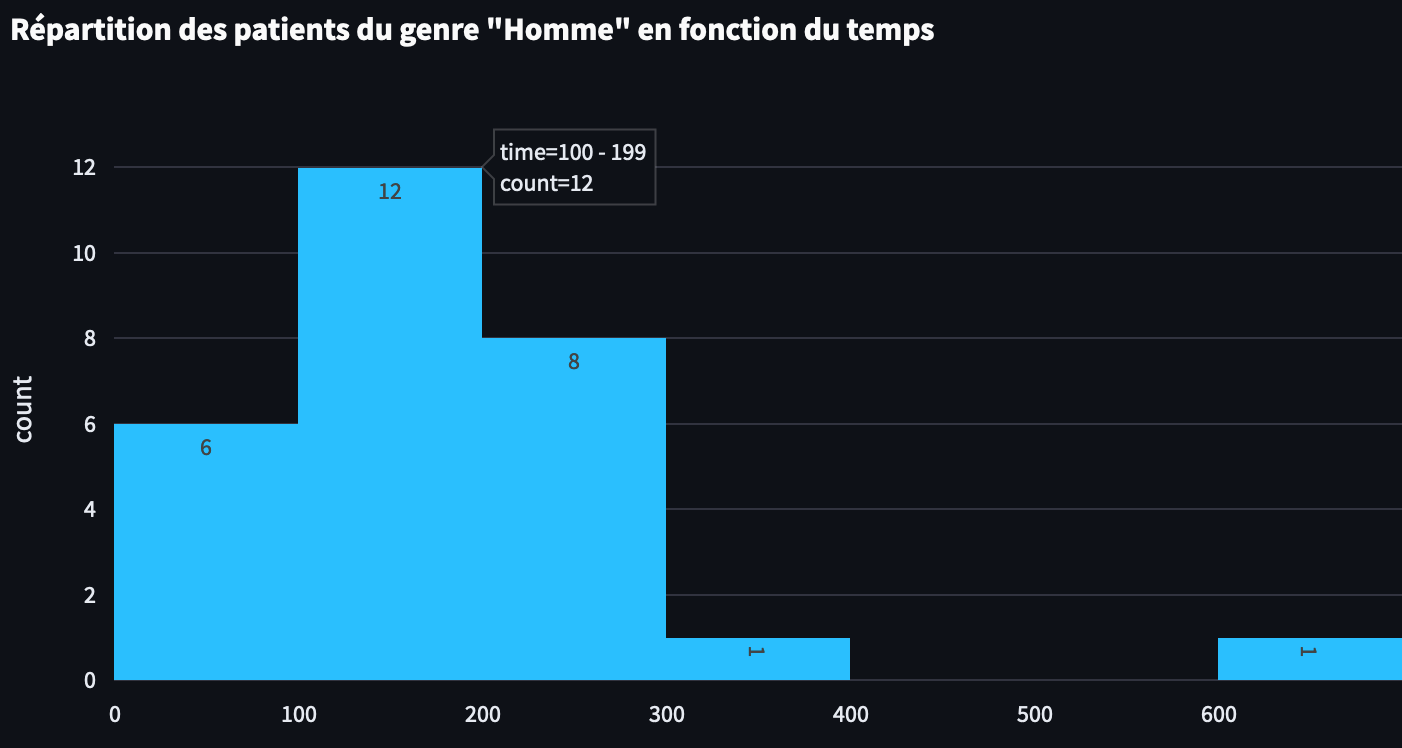
Présente les données transformées sous forme de table et avec une explication des transformations.

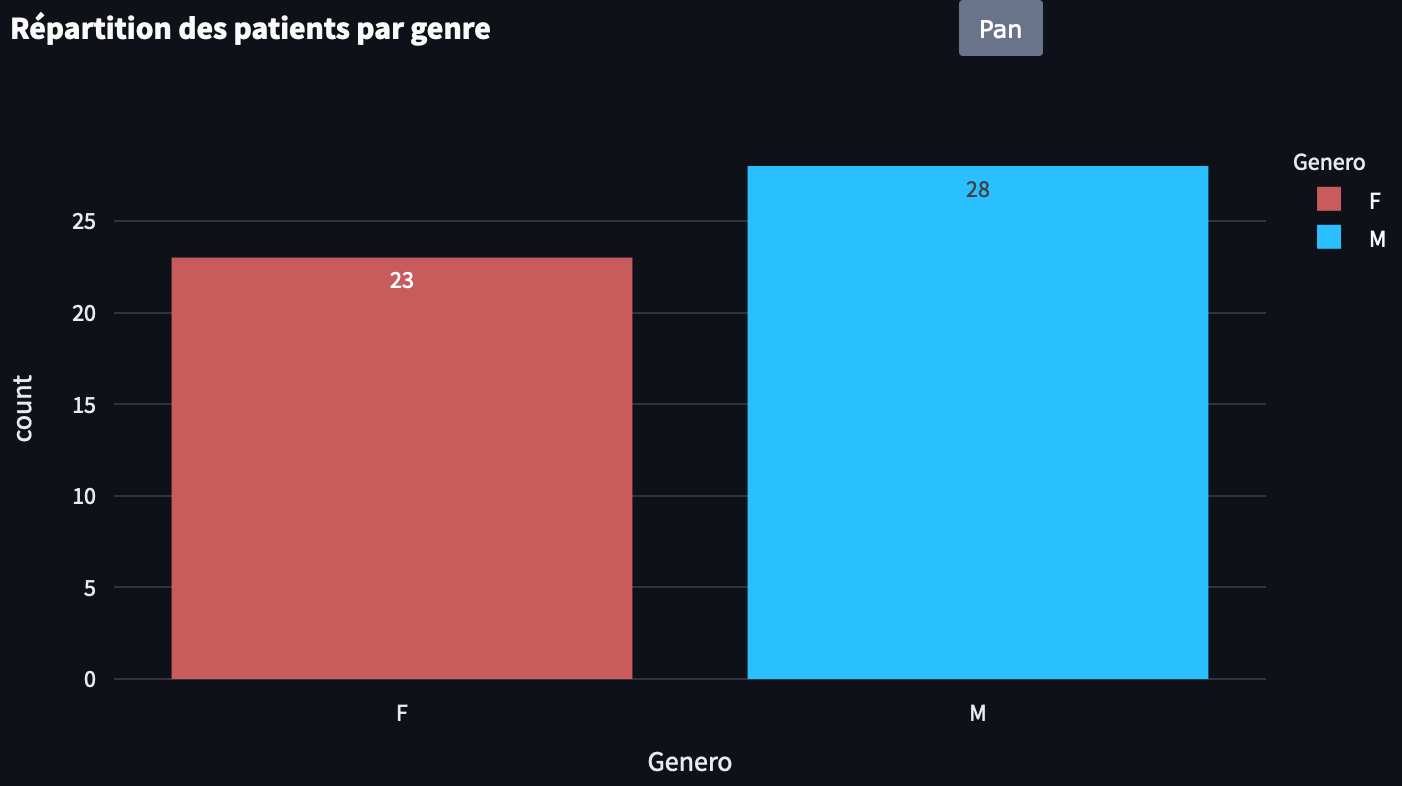
## Statistiques descriptives :

Cet onglet présente les variables sous forme statistique.

## Représentations graphiques des variables :

Représente sous forme graphique le nombre de patients en fonction du temps et du genre :

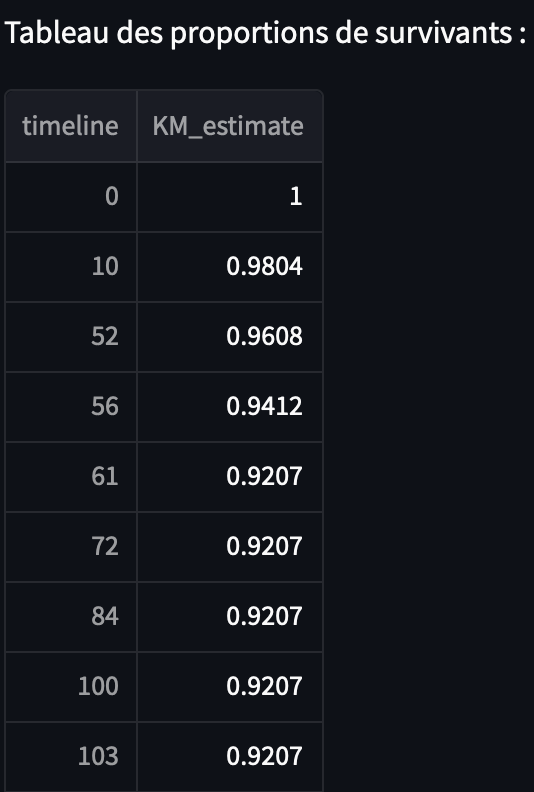
* Tous genres confondus  
  
* Féminin  
  
* Masculin  
  
* si la colonne « Genre » n’est pas choisie pour filtrer les données dans le menu latéral, alors une option apparaît :  
  

Représente sous forme graphique le nombre de patients par genre.  


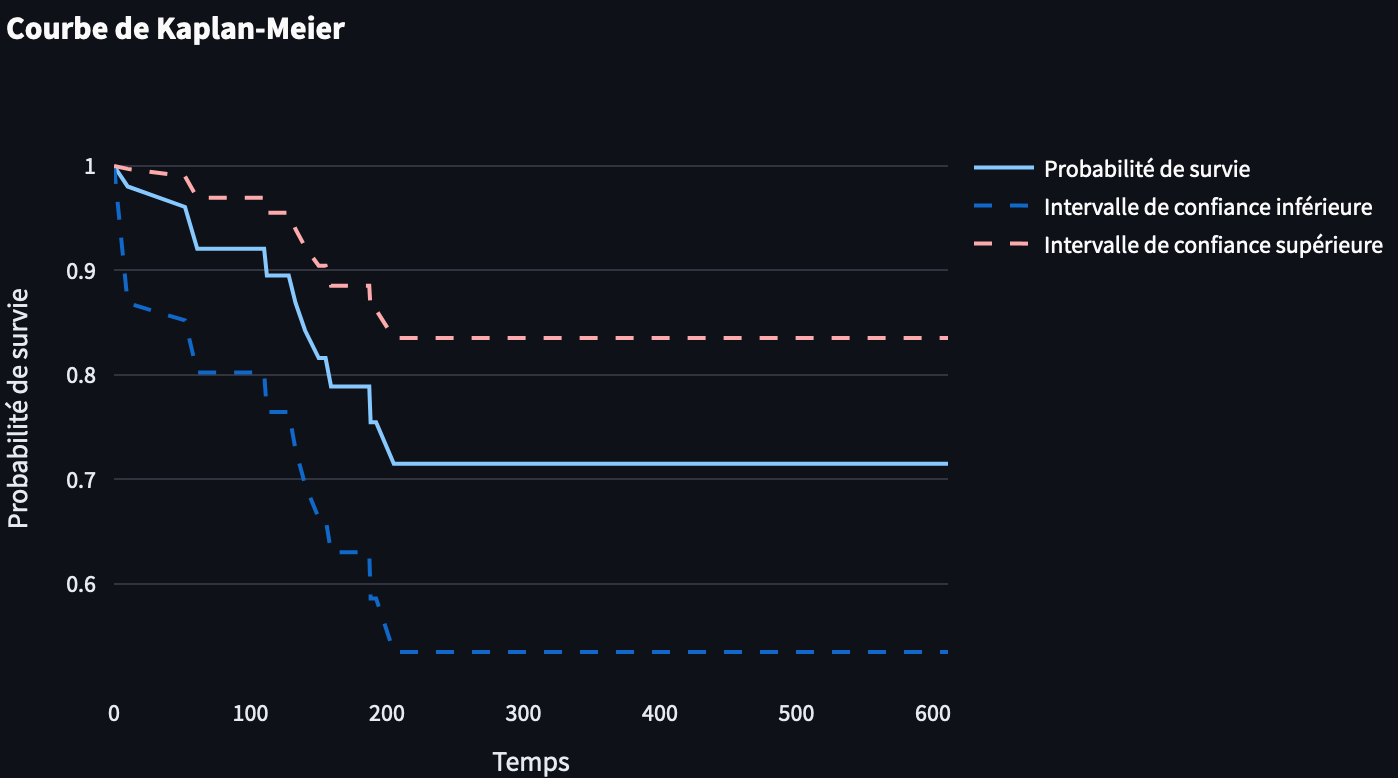
## Probabilités de survie et courbe de survie :

Dans cet onglet, nous avons 4 sujets proposés :

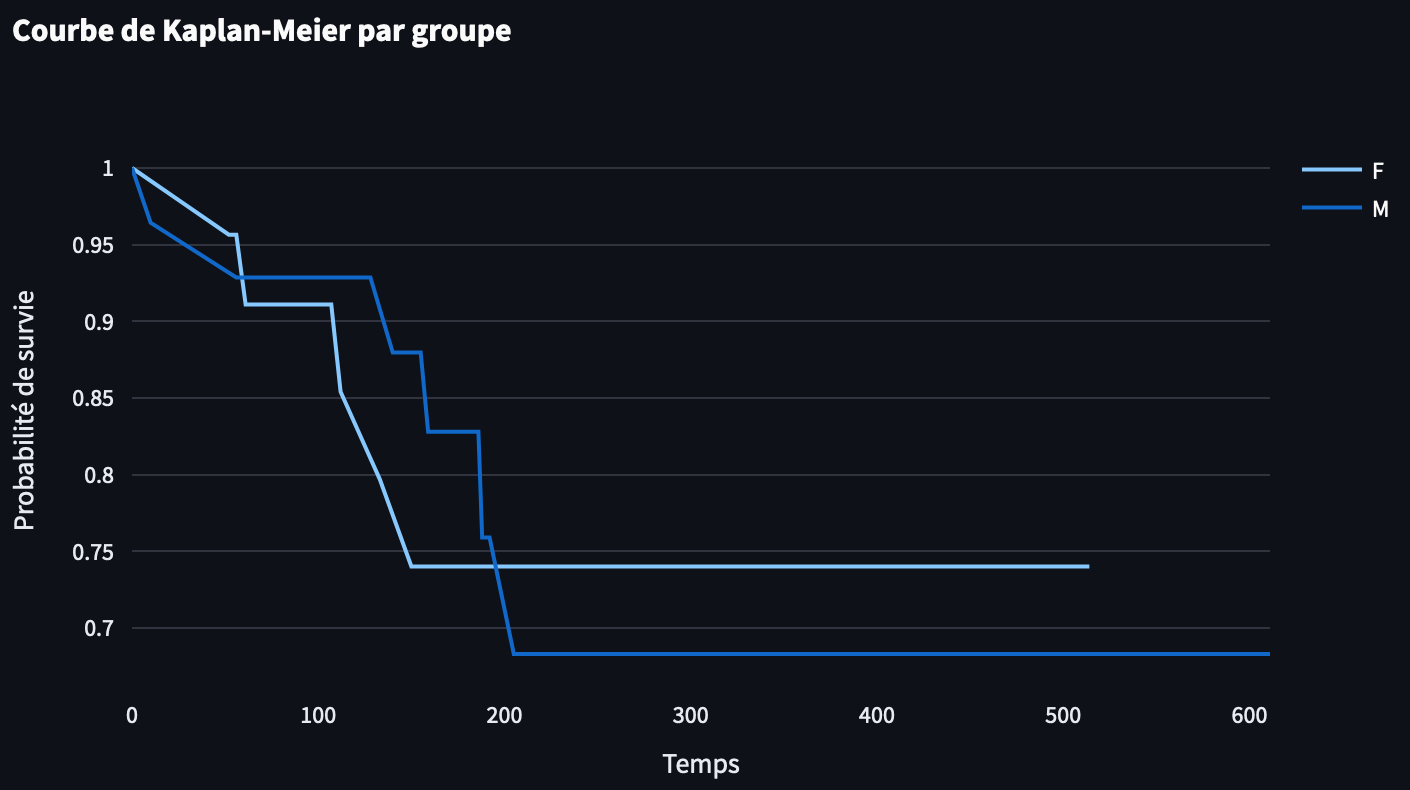
### Tableaux : Proportions de survivant et intervalle de confiance :

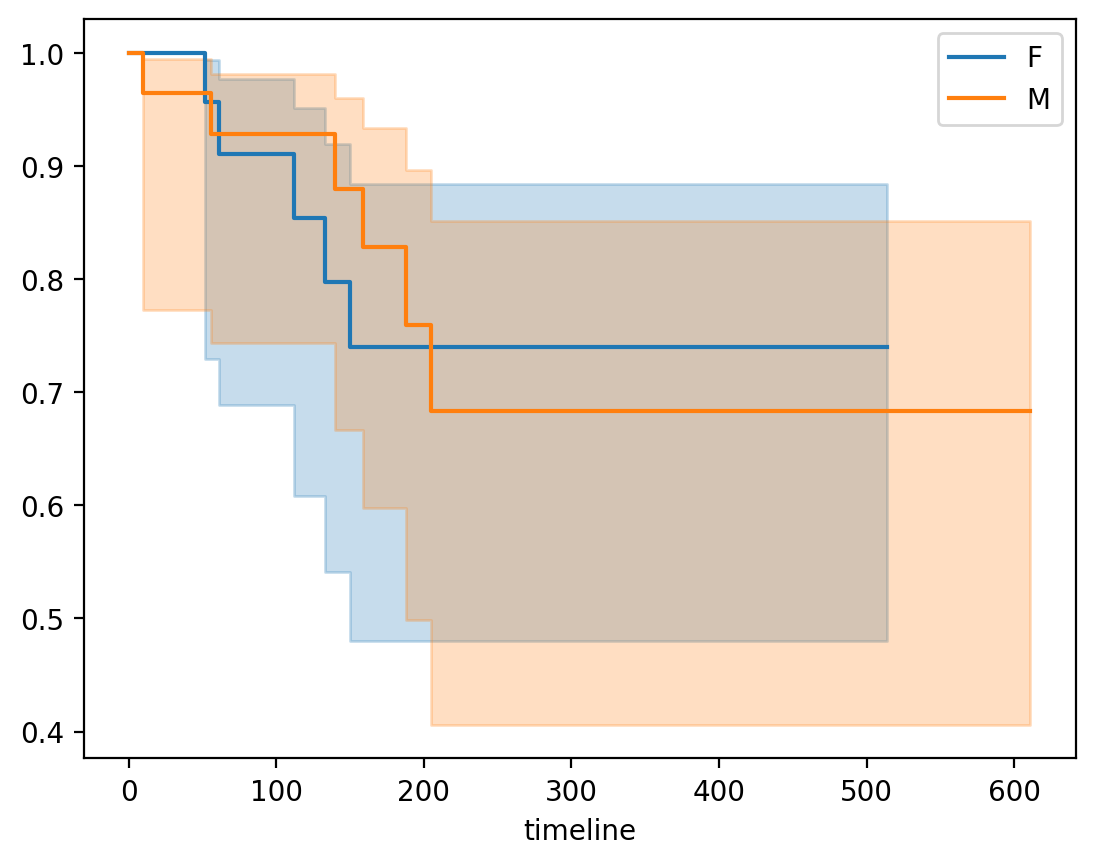
### Courbe de survie et intervalle de confiance :



### Courbe de survie par genre :



### Courbe de survie par genre avec intervalle de confiance :



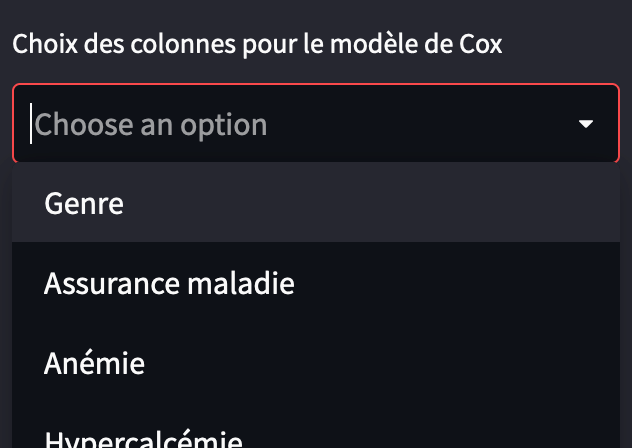
Créé avec matplotlib et non plotly qui ne pouvait pas générer ce visuel.

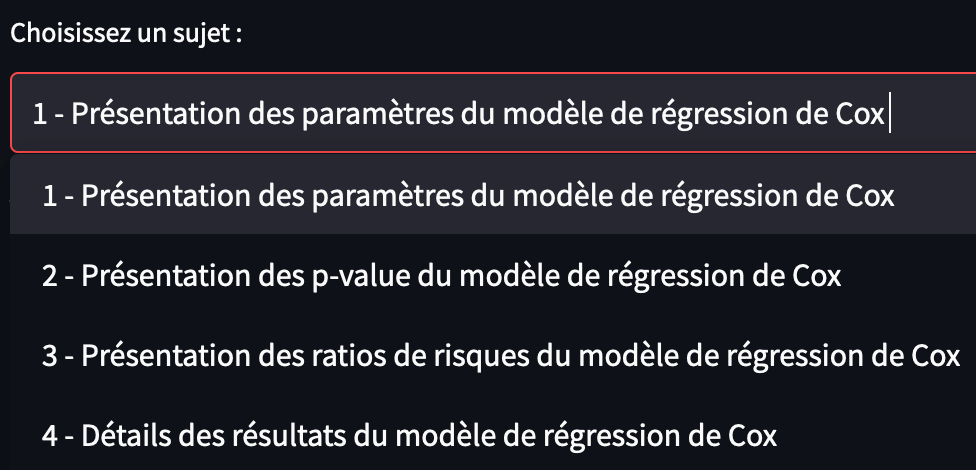
## Prédiction de survie d’un individu avec Kaplan-Meier :

Ici, l’utilisateur a la possibilité d’utiliser un slider pour indiquer une durée de vie et utiliser le modèle de Kaplan-Meier pour prédire la probabilité de survie à ce temps là.  

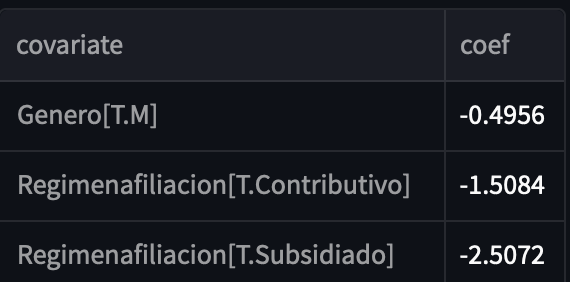

## Modèle de régression de Cox :

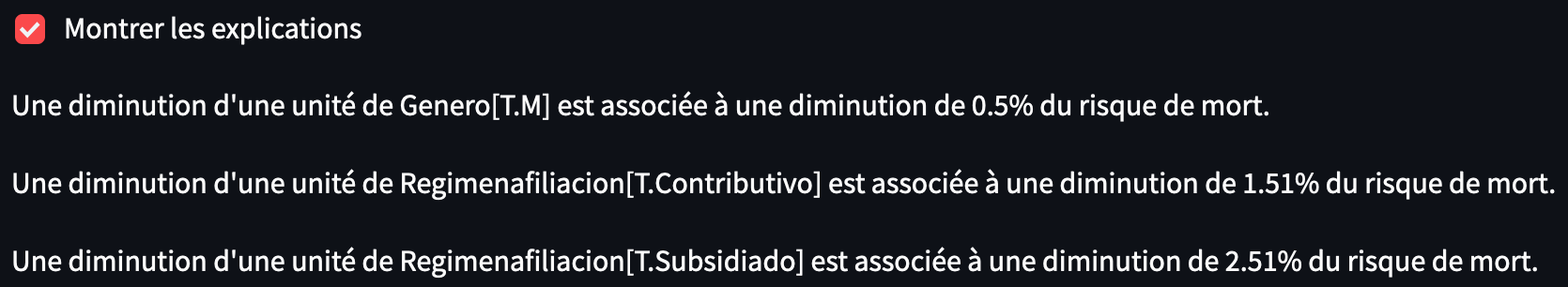
Après avoir choisi des colonnes dans le menu latéral il est possible de choisir 4 sujets :



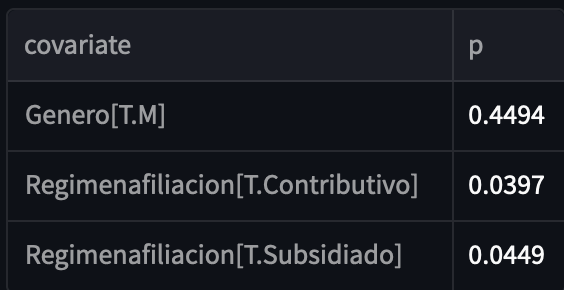


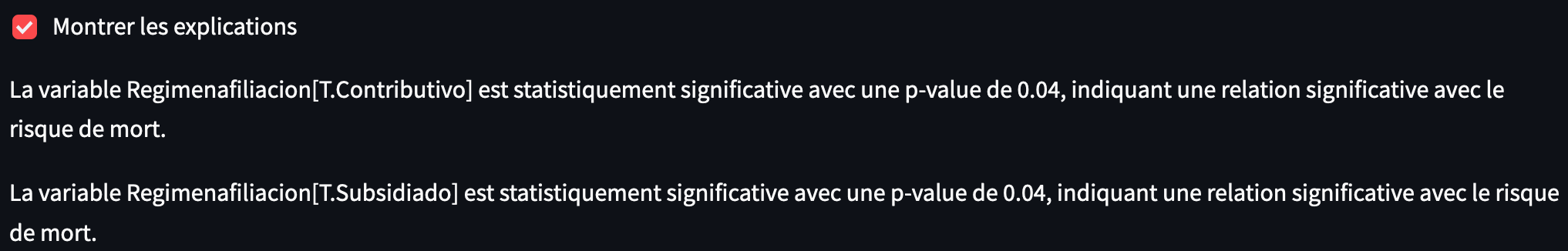
### Présentation des paramètres du modèle de régression de Cox :

Cette partie présente les paramètres du modèle de régression de Cox sous forme de tableau :  


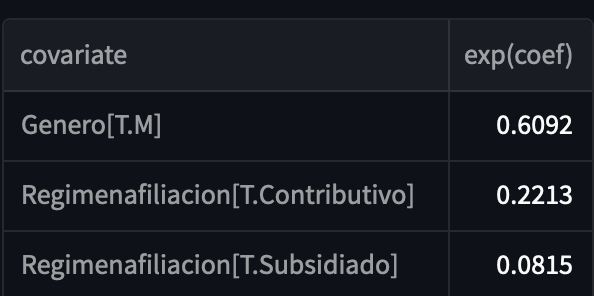
Des explications automatiques sont proposées :  


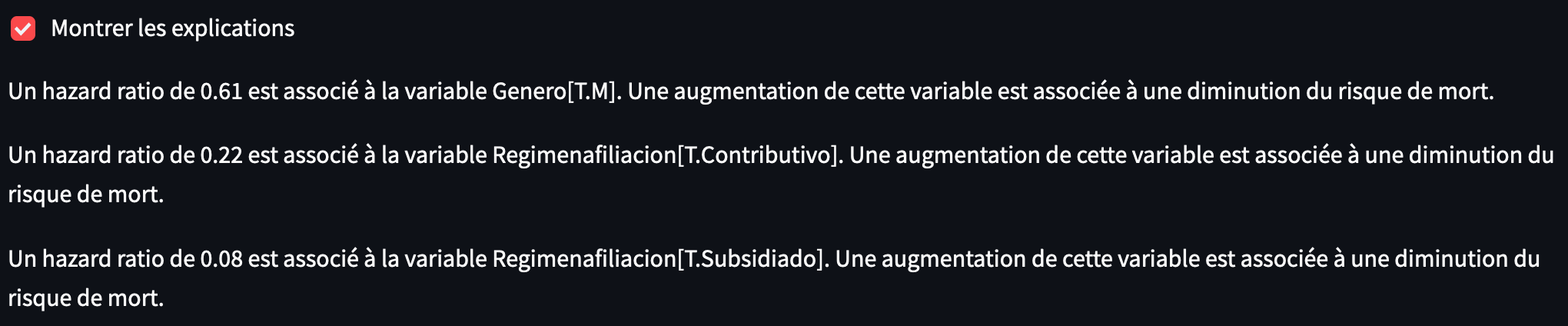
### Présentation des p-value du modèle de régression de Cox :

Cette partie présente les p-value du modèle de régression de Cox sous forme de tableau :  


Des explications automatiques sont proposées :  


### Présentation des ratios de risques du modèle de régression de Cox :

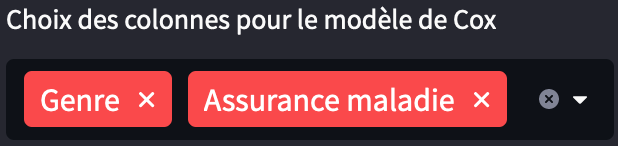
Cette partie présente les rations de risques sous forme de tableau :  


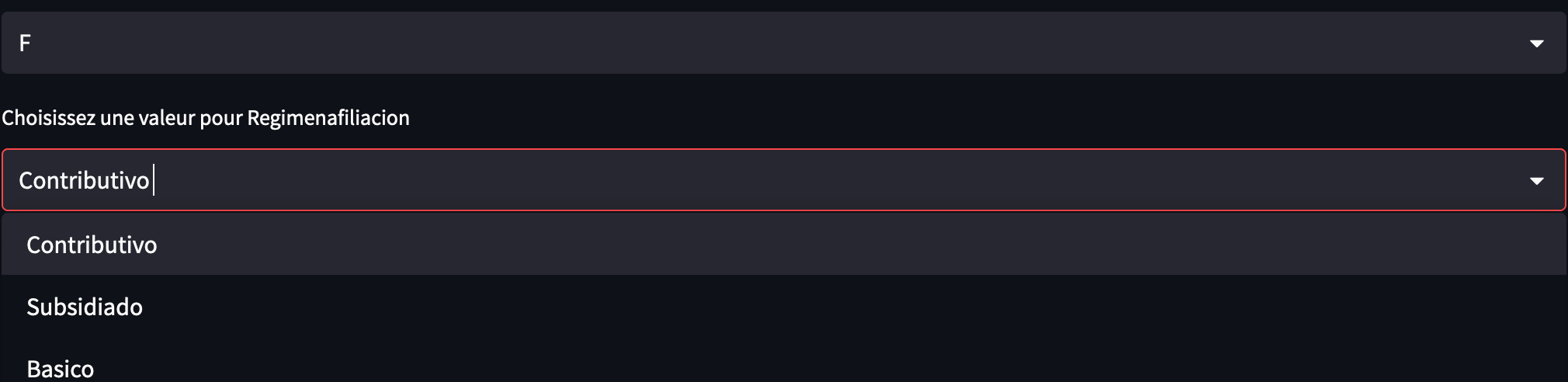
Des explications automatiques sont proposées :  


### Détails des résultats du modèle de régression de Cox :

Cette partie présente le résultat complet du modèle de régression de Cox :  


## Prédiction de survie d’un individu avec Cox :

Cet onglet propose à l’utilisateur de créé un individu ayant les mêmes colonnes que celles utilisés pour le modèle de régression de Cox (choisies dans le menu latéral).  




En fonction des choix choisis, le graphique suivant montre la courbe de survie de l’individu créé :  


## Analyse coût-efficacité :

Cet onglet était demandé dans le sujet du projet. Cependant, nous n’avons pas vu ce sujet en cours et n’est donc pas à faire.

# Décrire les possibilités d’hébergement web :

Nous avons fait le choix d’utiliser l’hébergement gratuit de Streamlit pour partager notre application.

Autrement, il faut un domaine et un serveur pouvant faire fonctionner des programmes python.

# Ce qui ne fonctionne pas :

Tout fonctionne.