Introduction :

Nous allons maintenant analyser notre dataset pour comprendre quelles sont les tendances principales. Cette étape est essentielle, car elle nous permet d'identifier les facteurs de risque des maladies cardiaques avant même d'entraîner un modèle de prédiction. Nous allons utiliser des statistiques et des graphiques pour visualiser les données.

Dans cette partie, nous allons préparer notre environnement et nos données pour l'analyse. D'abord, nous installons Seaborn, une bibliothèque qui nous aide à créer des graphiques. Ensuite, nous téléversons notre fichier de données dans Google Colab, le chargeons dans un DataFrame avec Pandas, et regardons les premières lignes pour vérifier que tout est correct.

Comme vous pouvez le voir, les 5 premières lignes sont bien affichées.

Voici un tableau avec les statistiques de base pour mieux comprendre notre jeu de données.

Codage

```
print(df.describe().T)
```

```
              count        mean         std   min     25%    50%    75%    max
Age           918.0   53.510893    9.432617  28.0   47.00   54.0   60.0   77.0
RestingBP     918.0  132.396514   18.514154   0.0  120.00  130.0  140.0  200.0
Cholesterol   918.0  198.799564  109.384145   0.0  173.25  223.0  267.0  603.0
FastingBS     918.0    0.233115    0.423046   0.0    0.00    0.0    0.0    1.0
MaxHR         918.0  136.809368   25.460334  60.0  120.00  138.0  156.0  202.0
Oldpeak       918.0    0.887364    1.066570  -2.6    0.00    0.6    1.5    6.2
HeartDisease  918.0    0.553377    0.497414   0.0    0.00    1.0    1.0    1.0
```

Ce tableau présente les **statistiques descriptives** des variables numériques du dataset. On y voit notamment le **nombre d'échantillons** (count), la **moyenne** (mean), l'**écart-type (écart à la moyenne)** (std), les **valeurs minimale et maximale**, ainsi que les **quartiles** (25 %, 50 %, 75 %). Cela permet de saisir rapidement la répartition de chaque variable (par exemple, l'âge, la tension artérielle au repos, le cholestérol, etc.) et de repérer d'éventuelles valeurs extrêmes.

Voici un bref aperçu des variables et de leurs statistiques (issues de `df.describe().T`) :

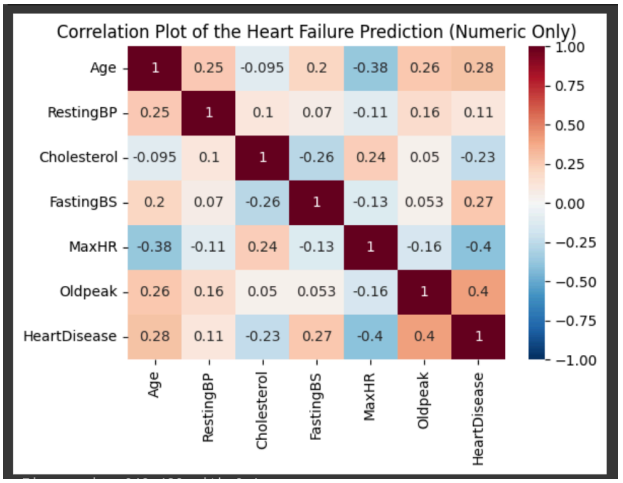
- **Age** : varie de 28 à 77 ans, avec une moyenne autour de 53 ans.
- **RestingBP (pression artérielle au repos)** : en moyenne 132 mmHg, mais on observe un minimum à 0 (probablement une valeur aberrante) et un maximum à 200.
- **Cholesterol** : moyenne proche de 199 mg/dl, allant jusqu'à 603 mg/dl, ce qui indique une large dispersion.
- **FastingBS (glycémie à jeun)** : la moyenne étant ~0,23, cela suggère qu'environ 23 % des individus ont une glycémie élevée (codée en 1).

- **MaxHR (fréquence cardiaque maximale)** : moyenne autour de 137 bpm, avec un minimum à 60 et un maximum à 202.
- **Oldpeak** : mesure de la dépression ST ; varie de 0 à 6,2, la moyenne étant inférieure à 1.
- **HeartDisease** : valeur binaire (0 ou 1), avec une moyenne d'environ 0,55, ce qui signifie qu'environ 55 % des individus du dataset souffrent de maladie cardiaque.

Ces statistiques permettent de cerner rapidement la **répartition** (minimum, maximum, moyenne, écart-type) de chaque variable et d'identifier de possibles valeurs inhabituelles (comme un RestingBP : rythme cardiaque au repos à 0).

Maintenant, examinons la **matrice de corrélation**, qui permet de visualiser les liens linéaires entre les différentes variables numériques de notre dataset.

Codage



Ce **diagramme de corrélation** montre, pour chaque paire de variables numériques, un **coefficient de corrélation** (entre -1 et +1) qui indique dans quelle mesure ces variables varient ensemble :

- **Valeur positive** (proche de +1) : plus l'une augmente, plus l'autre a tendance à augmenter.
- **Valeur négative** (proche de -1) : plus l'une augmente, plus l'autre a tendance à diminuer.
- **Valeur proche de 0** : pas de relation linéaire marquée entre les deux variables.

Les couleurs (allant du bleu/violet pour les corrélations négatives au rouge/orange pour les corrélations positives) aident à repérer rapidement les paires de variables les plus fortement corrélées. Dans le contexte de la **prédiction d'insuffisance cardiaque** (*Heart Failure Prediction*), on observe par exemple que :

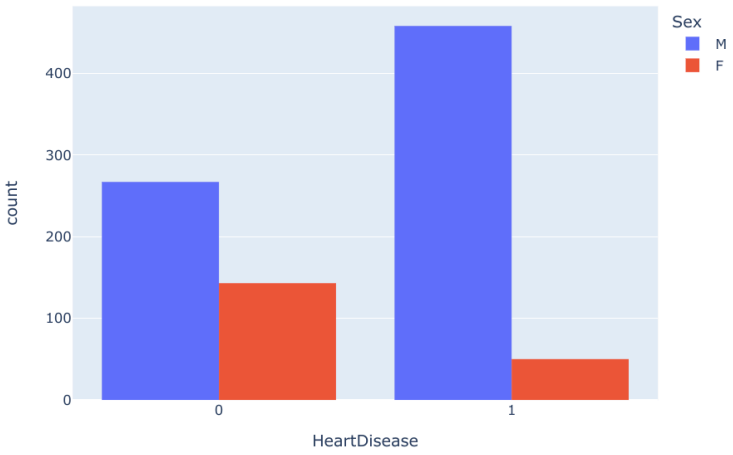
- **FastingBS** (glycémie à jeun), **Oldpeak** (dépression du segment ST) et **MaxHR** (fréquence cardiaque maximale) ont des corrélations plus marquées avec **HeartDisease** que d'autres variables (même si ces corrélations restent modérées).
- **Age** présente aussi une légère corrélation positive avec **HeartDisease**.
- **Cholesterol** et **RestingBP** semblent peu corrélés à la maladie, du moins selon cette mesure linéaire.

Cette heatmap permet donc d'identifier quelles variables pourraient être plus influentes dans la détection ou la prédiction d'une maladie cardiaque, même si une corrélation seule ne prouve pas de lien de cause à effet.

Gohar

Pour visualiser la répartition de la maladie cardiaque en fonction du sexe, nous allons créer un diagramme en barres comparant la variable **HeartDisease** (0 ou 1) à la variable **Sex** (M ou F).

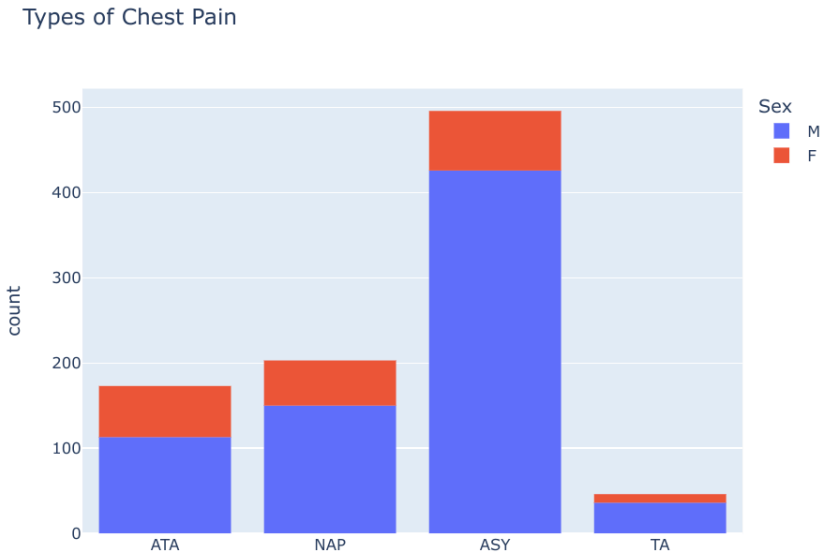
Codage



Dans notre étude, on constate qu'il y a plus d'hommes que de femmes (H=705; F=193). On constate que la différence entre les hommes et les femmes qui ont fait des crises cardiaques est très importante (1) alors que la différence entre ces deux sexes qui n'ont pas fait de crise cardiaque n'est pas très élevée. Le nombre d'hommes qui font des crises cardiaques est 2 fois plus important que ceux qui n'ont jamais eu de crise cardiaque. Alors que cette différence n'est pas très élevée chez les femmes.

Pour mieux comprendre la répartition des différents types de douleurs thoraciques (ChestPainType) selon le sexe, nous allons tracer un diagramme en barres permettant de visualiser la variable Sex au sein de chaque catégorie de ChestPainType.

Codage

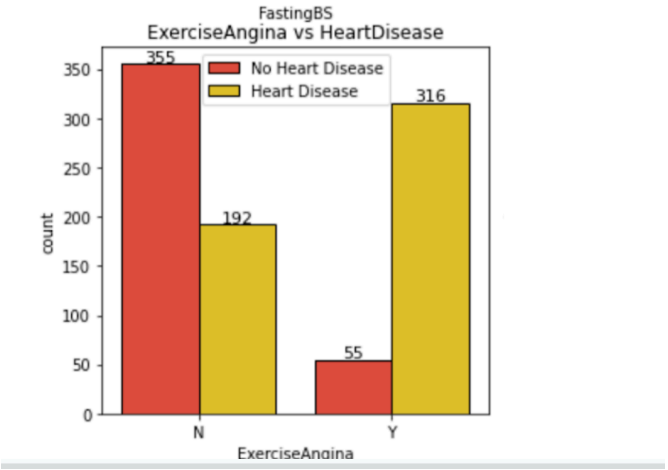


ATA (Atypical Angina) : Douleur thoracique atypique, moins caractéristique que l'angine classique, mais pouvant néanmoins être liée à une ischémie cardiaque.
NAP (Non-Anginal Pain) : Douleur thoracique non angineuse, qui n'est généralement pas provoquée par un problème cardiaque.
ASY (Asymptomatic) : Absence de douleur ou symptômes peu apparents, parfois appelée « ischémie silencieuse
TA (Typical Angina) : Douleur thoracique typique, souvent liée à l'effort et soulagée par le repos ou la nitroglycérine.

L'analyse des types de douleurs thoraciques révèle que la présence d'une douleur asymptomatique (ASY) est fortement associée aux cas d'insuffisance cardiaque. Même en l'absence de symptômes évidents, ce type de douleur pourrait indiquer une altération sous-jacente du fonctionnement cardiaque, faisant de ce critère un indicateur à surveiller de près.

Pour illustrer la relation entre l'angine d'effort (**ExerciseAngina**) et la présence d'une maladie cardiaque (**HeartDisease**), nous allons tracer un diagramme en barres groupées, mettant en évidence le rôle de la glycémie à jeun (**FastingBS**) dans l'interprétation des résultats.

codage



Ce diagramme en barres compare la présence ou l'absence de maladie cardiaque (**HeartDisease**) selon que le patient présente une angine d'effort (**ExerciseAngina**). Les barres rouges indiquent les individus sans maladie cardiaque, et les barres jaunes ceux atteints de maladie cardiaque. On observe ici que, parmi les personnes n'ayant pas d'angine d'effort (N), la majorité n'a pas de maladie cardiaque, tandis que chez celles ayant une angine d'effort (Y), on constate davantage de cas de maladie cardiaque. Les chiffres au-dessus de chaque barre correspondent au nombre d'individus dans chaque catégorie. Le titre fait référence à **FastingBS (fasting blood sugar)**, suggérant que l'analyse est considérée en lien avec la glycémie à jeun, même si cette variable n'est pas directement représentée sur l'axe du graphique.

Conclusion :

Cette analyse exploratoire souligne l'importance de variables comme l'âge, la glycémie à jeun (FastingBS), l'angine d'effort (ExerciseAngina) et la fréquence cardiaque maximale (MaxHR), tout en révélant des différences notables entre hommes et femmes. L'attention portée aux douleurs thoraciques asymptomatiques (ASY) et aux valeurs extrêmes oriente les prochaines étapes de préparation et de modélisation des données.

Exercice angina : désigne une douleur thoracique qui apparaît généralement pendant un effort ou un stress.

Pourquoi dans notre étude nous avons plus de d'hommes que de femmes ?

Prévalence des maladies cardiaques : Les maladies cardiaques sont souvent plus étudiées chez les hommes, car ils ont tendance à présenter des symptômes plus évidents et

sont plus fréquemment inclus dans les études médicales sur ce sujet.

Comportement en matière de santé : Les hommes consultent parfois plus tardivement que les femmes pour des problèmes cardiaques, ce qui pourrait les rendre plus représentés dans un échantillon lié à des diagnostics de maladies cardiovasculaires.

Angina pectoris: refers to chest pain that usually appears during exertion or stress.

Why do we have more men than women in our study?

Prevalence of heart disease: Heart disease is often studied more in men because they tend to have more obvious symptoms and are more frequently included in medical studies on this subject.

Health behaviour: Men sometimes consult later than women for heart problems, which could make them more represented in a sample linked to cardiovascular disease diagnoses.

ETAPE	CODE	RESULTAT
1	`!pip install seaborn`	`!pip install seaborn` permet donc de **télécharger et installer la bibliothèque Seaborn** dans l'environnement d'exécution de ton notebook, afin de bénéficier de ses fonctionnalités de visualisation de données. !pip install seaborn allows you to **download and install the Seaborn library** in your notebook's runtime environment, so you can benefit from its data visualisation features.
2	`from google.colab import files` `uploaded = files.upload()`	Dans un notebook Google Colab, ces deux lignes de code servent à **uploader** un fichier depuis ton ordinateur vers l'environnement Colab In a Google Colab notebook, these two lines of code are used to **upload** a file from your computer to the Colab environment

3	``` import pandas as pd df = pd.read_csv('heart.csv') print(df.head()) ```	Ce code charge les données du fichier `heart.csv` dans un DataFrame Pandas et affiche un aperçu des premières lignes pour s'assurer que tout est en ordre. This code loads the data from the heart.csv file into a Pandas DataFrame and displays an overview of the first lines to ensure that everything is in order.
4	``` print(df.describe().T) ```	**`df.describe()`** : affiche les statistiques descriptives de toutes les colonnes numériques (count, mean, std, min, quartiles, max). **`.T`** : transpose le résultat, ce qui place les noms de colonnes en index (à gauche) et les différentes statistiques en colonnes. **df.describe()**: displays descriptive statistics for all numeric columns (count, mean, std, min, quartiles, max). **.T**: transposes the result, which places the column names as indexes (on the left) and the different statistics in columns.
5	``` import pandas as pd import seaborn as sns import matplotlib.pyplot as plt df = pd.read_csv('heart.csv') df_numeric = pd.read_csv('heart.csv') df_numeric = df_numeric.select_dtypes(include=[np.number]) corr = df_numeric.corr() plt.figure(figsize=(6,4)) sns.heatmap(corr, annot=True, cmap='RdBu_r', vmin=-1, vmax=1) plt.title('Correlation Plot of the Heart Failure Prediction (Numeric Only)') plt.show() plt.clf() ```	En résumé, ce script lit le fichier "heart.csv", isole les colonnes numériques, calcule la corrélation entre elles et affiche cette matrice de corrélation sous forme de heatmap, facilitant ainsi l'analyse des relations linéaires entre les variables du dataset. In summary, this script reads the 'heart.csv' file, isolates the numerical columns, calculates the correlation between them and displays this correlation matrix in the form of a heatmap, thus facilitating the analysis of the linear relationships between the variables in the dataset.

6	``` if 'HeartDisease' in df.columns and 'Sex' in df.columns: plt.figure(figsize=(6,4)) sns.countplot(data=df, x='HeartDisease', hue='Sex') plt.title('HeartDisease by Sex') plt.xlabel('HeartDisease') plt.ylabel('Count') plt.show() plt.clf() ```	En résumé, ce code génère un graphique qui permet de comparer visuellement le nombre d'individus présentant ou non une maladie cardiaque (**HeartDisease**) en fonction de leur sexe (**Sex**). In short, this code generates a graph that allows you to visually compare the number of individuals with or without heart disease (**HeartDisease**) according to their sex (**Sex**).
7	``` if 'ChestPainType' in df.columns and 'Sex' in df.columns: ctab = pd.crosstab(df['ChestPainType'], df['Sex']) ctab.plot(kind='bar', stacked=True, figsize=(6,4)) plt.title('Types of Chest Pain (Stacked by Sex)') plt.xlabel('ChestPainType') plt.ylabel('Count') plt.show() plt.clf() ```	Ce code crée un tableau croisé qui compte le nombre d'occurrences de chaque type de douleur thoracique par sexe, puis génère un diagramme en barres empilées pour visualiser ces données. Cela te permet de comparer facilement la répartition des types de douleurs thoraciques entre les hommes et les femmes. This code creates a cross-tabulation that counts the number of occurrences of each type of chest pain by sex, and then generates a stacked bar chart to visualise this data. This allows you to easily compare the distribution of types of chest pain between men and women.
8	``` if 'HeartDisease' in df.columns: df['HeartDisease'] = df['HeartDisease'].map({0: 'No Heart Disease', 1: 'Heart Disease'}) plt.figure(figsize=(6,4)) ax = sns.countplot(data=df, x='ExerciseAngina', hue='HeartDisease', palette={'No Heart Disease': 'red', 'Heart Disease': 'gold'}) ```	Ce code transforme d'abord la colonne **HeartDisease** en étiquettes explicites ("No Heart Disease" et "Heart Disease"). Ensuite, il crée un graphique en barres groupées qui affiche la répartition de la variable **ExerciseAngina** en fonction de **HeartDisease** et ajoute les valeurs de comptage sur chaque barre. Enfin, il personnalise les titres et les labels des axes avant d'afficher le graphique final. This code first transforms the **HeartDisease** column into explicit labels ('No Heart Disease' and 'Heart Disease'). Then it creates a grouped bar chart that displays the distribution of the **ExerciseAngina** variable as a function of **HeartDisease** and adds the count values to each

	`` for container in ax.containers: ax.bar_label(container) plt.suptitle('FastingBS', fontsize=12) plt.title('ExerciseAngina vs HeartDisease', fontsize=10) plt.xlabel('ExerciseAngina') plt.ylabel('Count') plt.show() ```	bar. Finally0108, it customises the axis titles and labels before displaying the final chart.