

# Titanic Seaborn Visualization

June 21, 2024

## 0.1 Importation des packages

```
[ ]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

## 0.2 Chargement des datas

```
[2]: df = pd.read_csv("C:/Users/JEHDAl/Desktop/Python Practice/Project/Titanic_
↳Seaborn Visualisation/Dataset/titanic.csv")

df.head(50)
```

```
[2]:
```

|    | survived | pclass | sex    | age  | sibsp | parch | fare    | embarked | class  | \ |
|----|----------|--------|--------|------|-------|-------|---------|----------|--------|---|
| 0  | 0        | 3      | male   | 22.0 | 1     | 0     | 7.2500  | S        | Third  |   |
| 1  | 1        | 1      | female | 38.0 | 1     | 0     | 71.2833 | C        | First  |   |
| 2  | 1        | 3      | female | 26.0 | 0     | 0     | 7.9250  | S        | Third  |   |
| 3  | 1        | 1      | female | 35.0 | 1     | 0     | 53.1000 | S        | First  |   |
| 4  | 0        | 3      | male   | 35.0 | 0     | 0     | 8.0500  | S        | Third  |   |
| 5  | 0        | 3      | male   | NaN  | 0     | 0     | 8.4583  | Q        | Third  |   |
| 6  | 0        | 1      | male   | 54.0 | 0     | 0     | 51.8625 | S        | First  |   |
| 7  | 0        | 3      | male   | 2.0  | 3     | 1     | 21.0750 | S        | Third  |   |
| 8  | 1        | 3      | female | 27.0 | 0     | 2     | 11.1333 | S        | Third  |   |
| 9  | 1        | 2      | female | 14.0 | 1     | 0     | 30.0708 | C        | Second |   |
| 10 | 1        | 3      | female | 4.0  | 1     | 1     | 16.7000 | S        | Third  |   |
| 11 | 1        | 1      | female | 58.0 | 0     | 0     | 26.5500 | S        | First  |   |
| 12 | 0        | 3      | male   | 20.0 | 0     | 0     | 8.0500  | S        | Third  |   |
| 13 | 0        | 3      | male   | 39.0 | 1     | 5     | 31.2750 | S        | Third  |   |
| 14 | 0        | 3      | female | 14.0 | 0     | 0     | 7.8542  | S        | Third  |   |
| 15 | 1        | 2      | female | 55.0 | 0     | 0     | 16.0000 | S        | Second |   |
| 16 | 0        | 3      | male   | 2.0  | 4     | 1     | 29.1250 | Q        | Third  |   |
| 17 | 1        | 2      | male   | NaN  | 0     | 0     | 13.0000 | S        | Second |   |
| 18 | 0        | 3      | female | 31.0 | 1     | 0     | 18.0000 | S        | Third  |   |
| 19 | 1        | 3      | female | NaN  | 0     | 0     | 7.2250  | C        | Third  |   |
| 20 | 0        | 2      | male   | 35.0 | 0     | 0     | 26.0000 | S        | Second |   |
| 21 | 1        | 2      | male   | 34.0 | 0     | 0     | 13.0000 | S        | Second |   |

|    |   |   |        |      |   |   |          |   |        |
|----|---|---|--------|------|---|---|----------|---|--------|
| 22 | 1 | 3 | female | 15.0 | 0 | 0 | 8.0292   | Q | Third  |
| 23 | 1 | 1 | male   | 28.0 | 0 | 0 | 35.5000  | S | First  |
| 24 | 0 | 3 | female | 8.0  | 3 | 1 | 21.0750  | S | Third  |
| 25 | 1 | 3 | female | 38.0 | 1 | 5 | 31.3875  | S | Third  |
| 26 | 0 | 3 | male   | NaN  | 0 | 0 | 7.2250   | C | Third  |
| 27 | 0 | 1 | male   | 19.0 | 3 | 2 | 263.0000 | S | First  |
| 28 | 1 | 3 | female | NaN  | 0 | 0 | 7.8792   | Q | Third  |
| 29 | 0 | 3 | male   | NaN  | 0 | 0 | 7.8958   | S | Third  |
| 30 | 0 | 1 | male   | 40.0 | 0 | 0 | 27.7208  | C | First  |
| 31 | 1 | 1 | female | NaN  | 1 | 0 | 146.5208 | C | First  |
| 32 | 1 | 3 | female | NaN  | 0 | 0 | 7.7500   | Q | Third  |
| 33 | 0 | 2 | male   | 66.0 | 0 | 0 | 10.5000  | S | Second |
| 34 | 0 | 1 | male   | 28.0 | 1 | 0 | 82.1708  | C | First  |
| 35 | 0 | 1 | male   | 42.0 | 1 | 0 | 52.0000  | S | First  |
| 36 | 1 | 3 | male   | NaN  | 0 | 0 | 7.2292   | C | Third  |
| 37 | 0 | 3 | male   | 21.0 | 0 | 0 | 8.0500   | S | Third  |
| 38 | 0 | 3 | female | 18.0 | 2 | 0 | 18.0000  | S | Third  |
| 39 | 1 | 3 | female | 14.0 | 1 | 0 | 11.2417  | C | Third  |
| 40 | 0 | 3 | female | 40.0 | 1 | 0 | 9.4750   | S | Third  |
| 41 | 0 | 2 | female | 27.0 | 1 | 0 | 21.0000  | S | Second |
| 42 | 0 | 3 | male   | NaN  | 0 | 0 | 7.8958   | C | Third  |
| 43 | 1 | 2 | female | 3.0  | 1 | 2 | 41.5792  | C | Second |
| 44 | 1 | 3 | female | 19.0 | 0 | 0 | 7.8792   | Q | Third  |
| 45 | 0 | 3 | male   | NaN  | 0 | 0 | 8.0500   | S | Third  |
| 46 | 0 | 3 | male   | NaN  | 1 | 0 | 15.5000  | Q | Third  |
| 47 | 1 | 3 | female | NaN  | 0 | 0 | 7.7500   | Q | Third  |
| 48 | 0 | 3 | male   | NaN  | 2 | 0 | 21.6792  | C | Third  |
| 49 | 0 | 3 | female | 18.0 | 1 | 0 | 17.8000  | S | Third  |

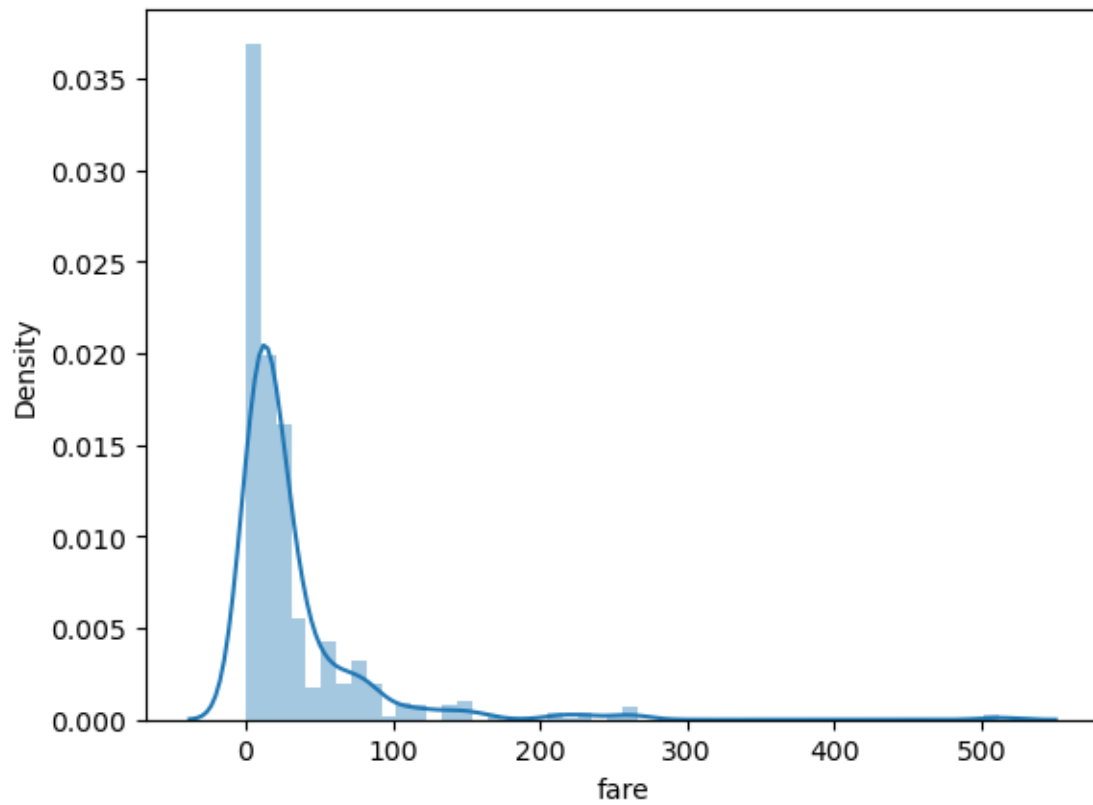
|    | who   | adult_male | deck | embark_town | alive | alone |
|----|-------|------------|------|-------------|-------|-------|
| 0  | man   | True       | NaN  | Southampton | no    | False |
| 1  | woman | False      | C    | Cherbourg   | yes   | False |
| 2  | woman | False      | NaN  | Southampton | yes   | True  |
| 3  | woman | False      | C    | Southampton | yes   | False |
| 4  | man   | True       | NaN  | Southampton | no    | True  |
| 5  | man   | True       | NaN  | Queenstown  | no    | True  |
| 6  | man   | True       | E    | Southampton | no    | True  |
| 7  | child | False      | NaN  | Southampton | no    | False |
| 8  | woman | False      | NaN  | Southampton | yes   | False |
| 9  | child | False      | NaN  | Cherbourg   | yes   | False |
| 10 | child | False      | G    | Southampton | yes   | False |
| 11 | woman | False      | C    | Southampton | yes   | True  |
| 12 | man   | True       | NaN  | Southampton | no    | True  |
| 13 | man   | True       | NaN  | Southampton | no    | False |
| 14 | child | False      | NaN  | Southampton | no    | True  |
| 15 | woman | False      | NaN  | Southampton | yes   | True  |
| 16 | child | False      | NaN  | Queenstown  | no    | False |

|    |       |       |     |             |     |       |
|----|-------|-------|-----|-------------|-----|-------|
| 17 | man   | True  | NaN | Southampton | yes | True  |
| 18 | woman | False | NaN | Southampton | no  | False |
| 19 | woman | False | NaN | Cherbourg   | yes | True  |
| 20 | man   | True  | NaN | Southampton | no  | True  |
| 21 | man   | True  | D   | Southampton | yes | True  |
| 22 | child | False | NaN | Queenstown  | yes | True  |
| 23 | man   | True  | A   | Southampton | yes | True  |
| 24 | child | False | NaN | Southampton | no  | False |
| 25 | woman | False | NaN | Southampton | yes | False |
| 26 | man   | True  | NaN | Cherbourg   | no  | True  |
| 27 | man   | True  | C   | Southampton | no  | False |
| 28 | woman | False | NaN | Queenstown  | yes | True  |
| 29 | man   | True  | NaN | Southampton | no  | True  |
| 30 | man   | True  | NaN | Cherbourg   | no  | True  |
| 31 | woman | False | B   | Cherbourg   | yes | False |
| 32 | woman | False | NaN | Queenstown  | yes | True  |
| 33 | man   | True  | NaN | Southampton | no  | True  |
| 34 | man   | True  | NaN | Cherbourg   | no  | False |
| 35 | man   | True  | NaN | Southampton | no  | False |
| 36 | man   | True  | NaN | Cherbourg   | yes | True  |
| 37 | man   | True  | NaN | Southampton | no  | True  |
| 38 | woman | False | NaN | Southampton | no  | False |
| 39 | child | False | NaN | Cherbourg   | yes | False |
| 40 | woman | False | NaN | Southampton | no  | False |
| 41 | woman | False | NaN | Southampton | no  | False |
| 42 | man   | True  | NaN | Cherbourg   | no  | True  |
| 43 | child | False | NaN | Cherbourg   | yes | False |
| 44 | woman | False | NaN | Queenstown  | yes | True  |
| 45 | man   | True  | NaN | Southampton | no  | True  |
| 46 | man   | True  | NaN | Queenstown  | no  | False |
| 47 | woman | False | NaN | Queenstown  | yes | True  |
| 48 | man   | True  | NaN | Cherbourg   | no  | False |
| 49 | woman | False | NaN | Southampton | no  | False |

**1. Dist plot Pour l'histogramme et voir la distribution d'une variable** Le `distplot()` montre la distribution de l'histogramme des données pour une seule colonne. Le nom de la colonne est passé en paramètre à la fonction `distplot()`

```
[3]: sns.distplot(df['fare'])
```

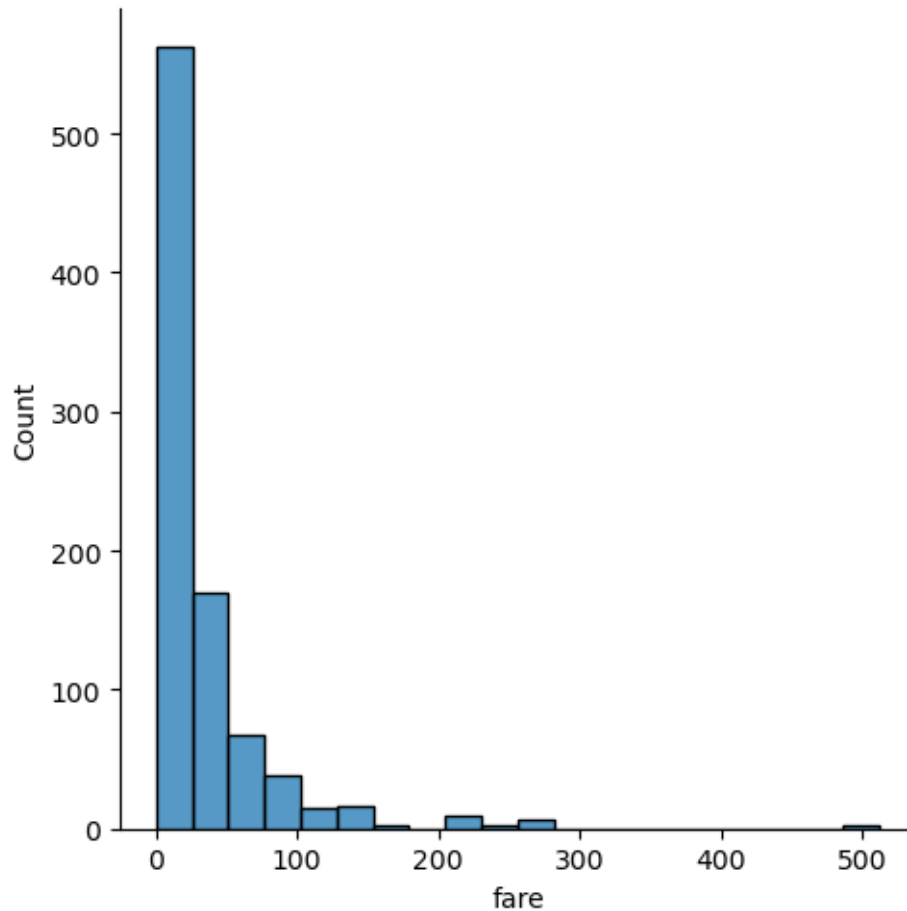
```
[3]: <Axes: xlabel='fare', ylabel='Density'>
```



Desactiver le kde

```
[8]: sns.displot(df['fare'], kde = False, bins = 20)
```

```
[8]: <seaborn.axisgrid.FacetGrid at 0x1cfacf97c10>
```

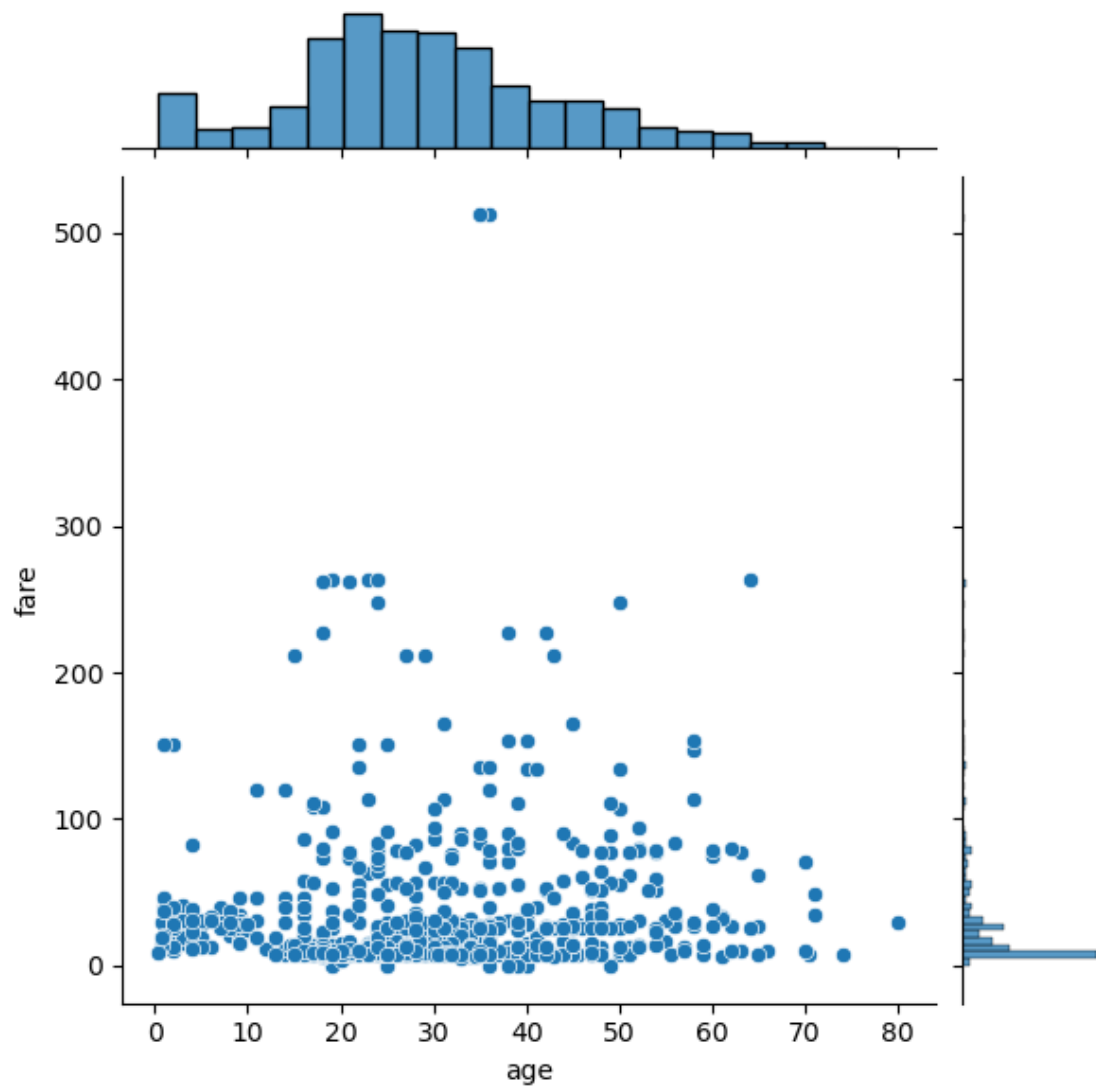


**2. JointPlot pour la corrélation entre deux variables** Le tracé conjoint est une technique de visualisation puissante dans Seaborn qui combine deux tracés différents en une seule visualisation : un nuage de points et un histogramme. Le nuage de points montre la relation entre deux variables, tandis que l'histogramme montre la distribution de chaque variable individuelle. Cela permet une analyse plus complète des données, car elle montre la corrélation entre les deux variables et leurs distributions individuelles.

```
[9]: sns.jointplot(x = 'age', y = 'fare', data = df)

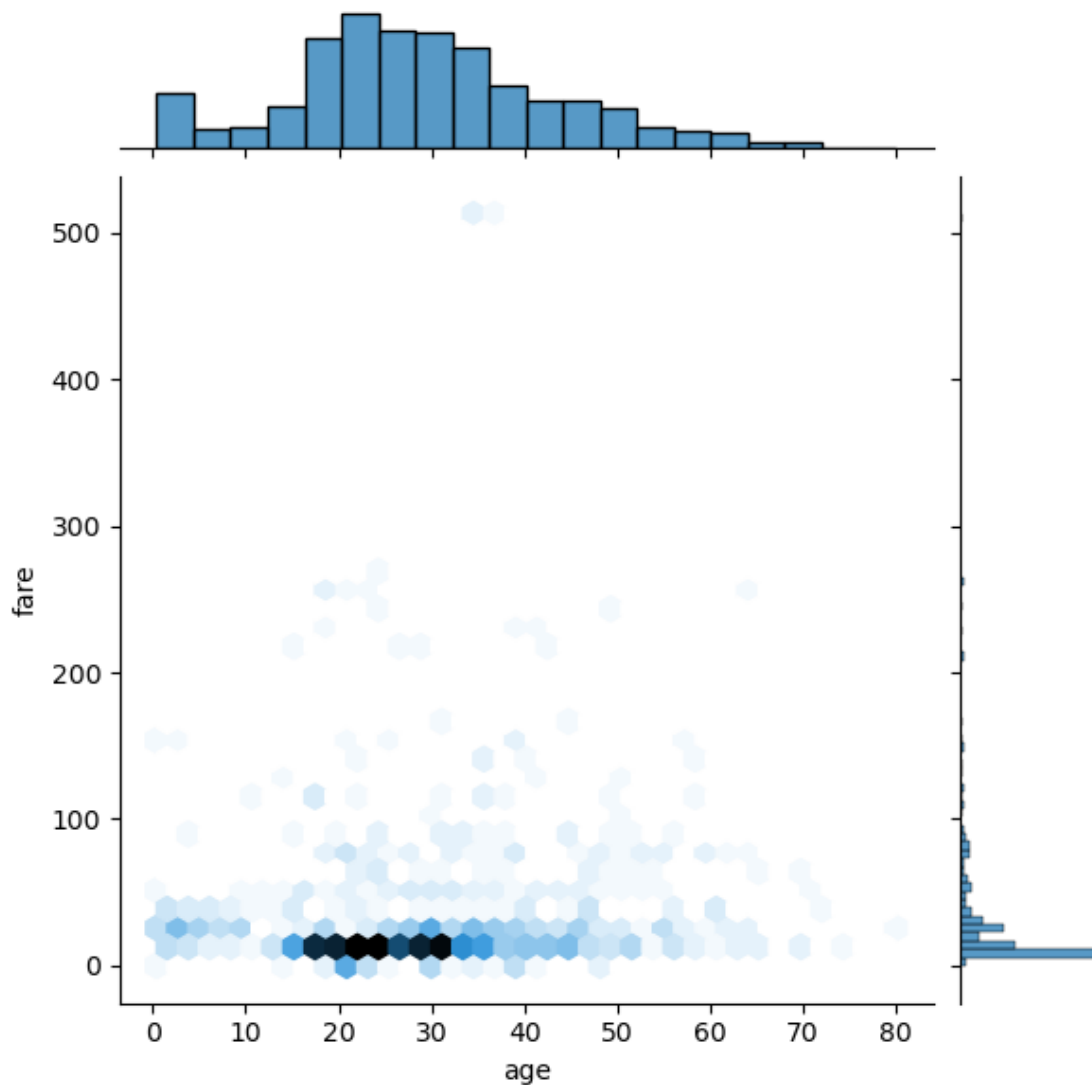
#Nous remarquons qu'il n'y a aucune corrélation entre les deux variables (age, fare)
```

```
[9]: <seaborn.axisgrid.JointGrid at 0x1cfad01e290>
```



```
[10]: sns.jointplot(x = "age", y = "fare", data = df, kind = "hex")
```

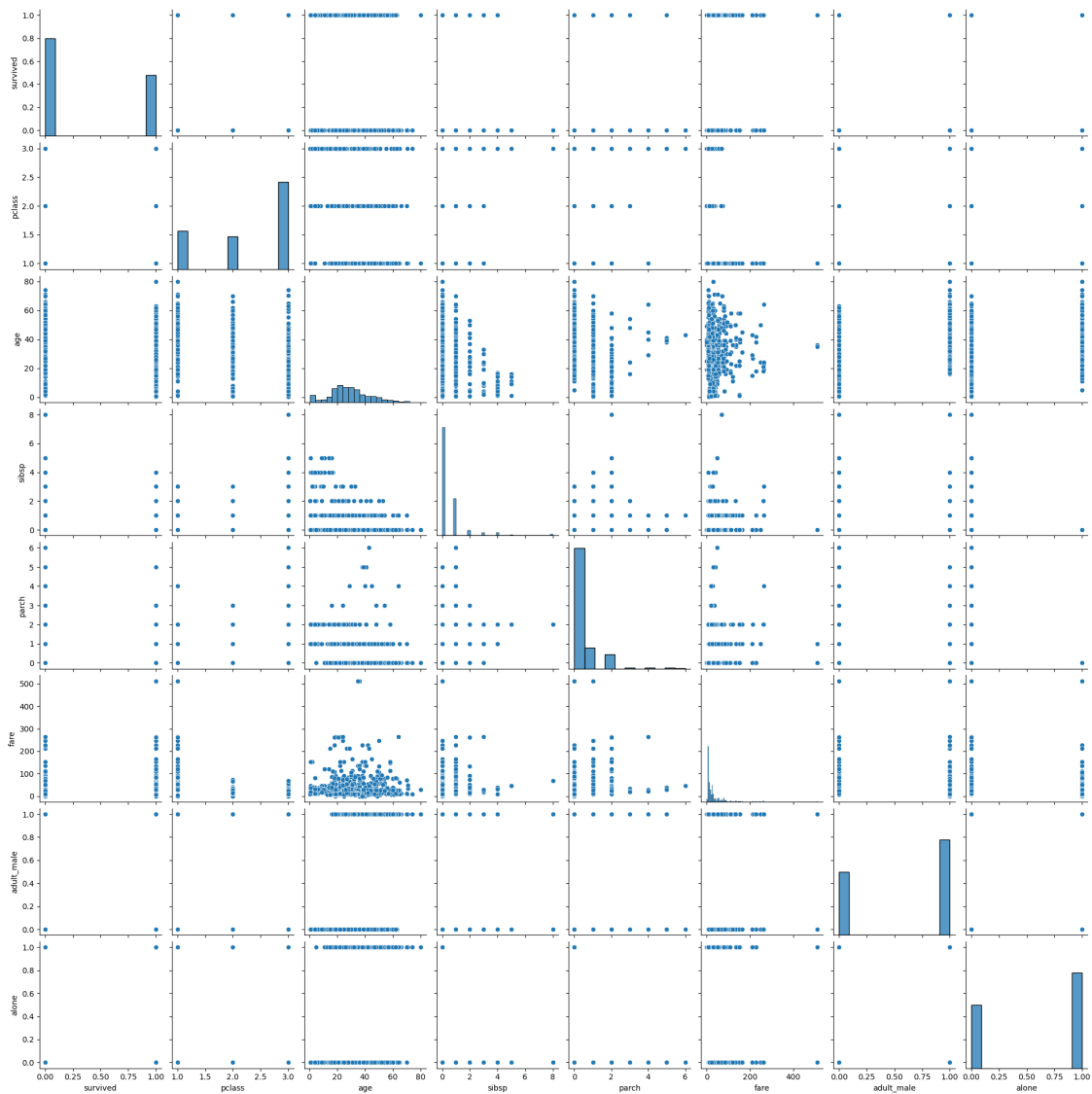
```
[10]: <seaborn.axisgrid.JointGrid at 0x1cf3e3e3a90>
```



**3. PairPlot (Pour une combinaison d'analyses de toutes les variables quantitatives du dataset pour voir les différentes relations-correlations entre deux variables et la diagonale est la relation-correlation entre la mm variable)** Les diagrammes de paires sont un type de visualisation dans lequel plusieurs nuages de points par paires sont affichés sous forme matricielle. Chaque nuage de points montre la relation entre deux variables, tandis que les diagrammes diagonaux montrent la distribution des variables individuelles.

```
[11]: sns.pairplot(df)
```

```
[11]: <seaborn.axisgrid.PairGrid at 0x1cf8e8f6e00>
```

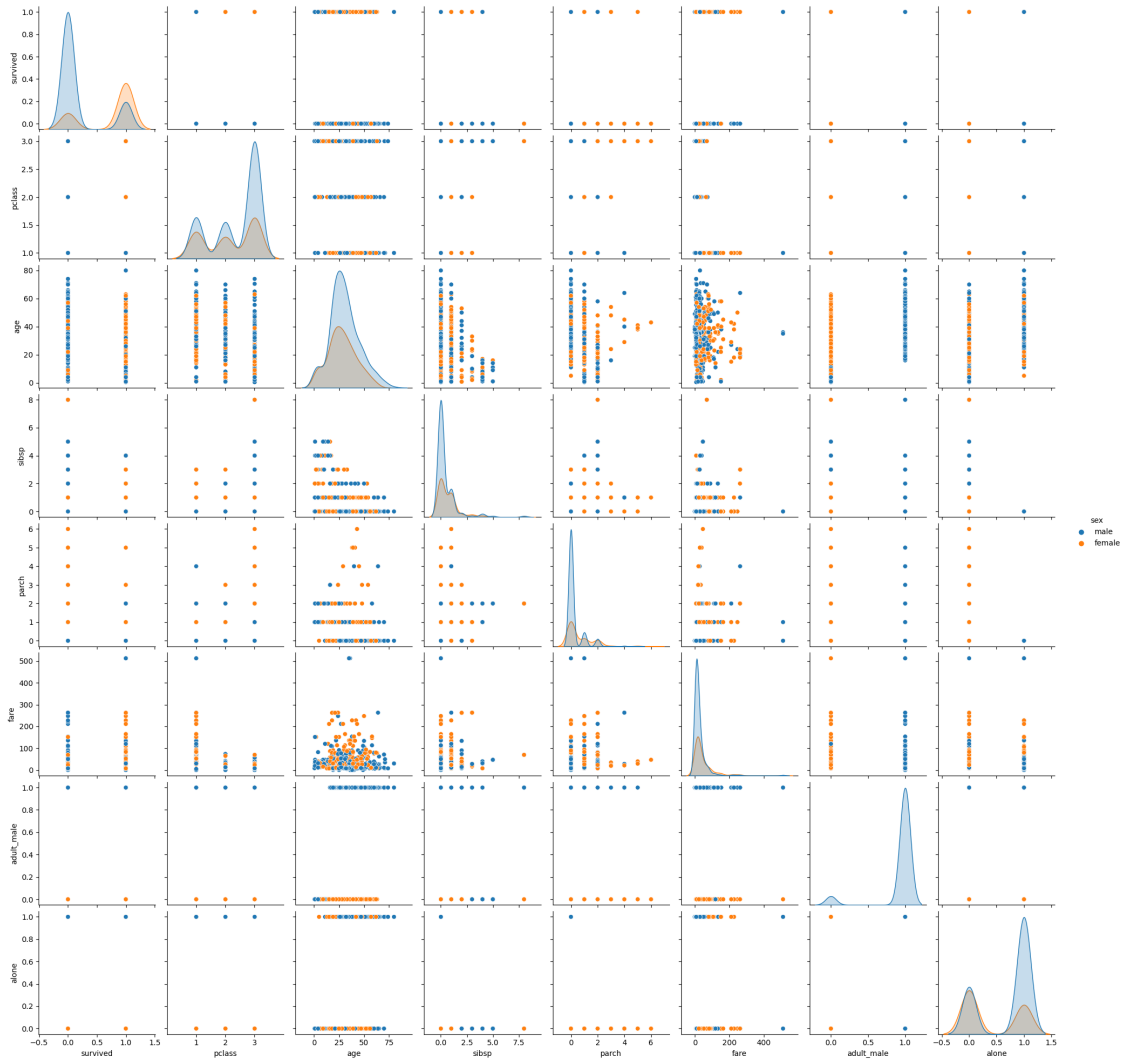


Ajouter une categorisation de pariplot analyz avec une autre variable

```
[12]: sns.pairplot(data = df, hue = 'sex')
```

```
[12]: <seaborn.axisgrid.PairGrid at 0x1cfb2328100>
```



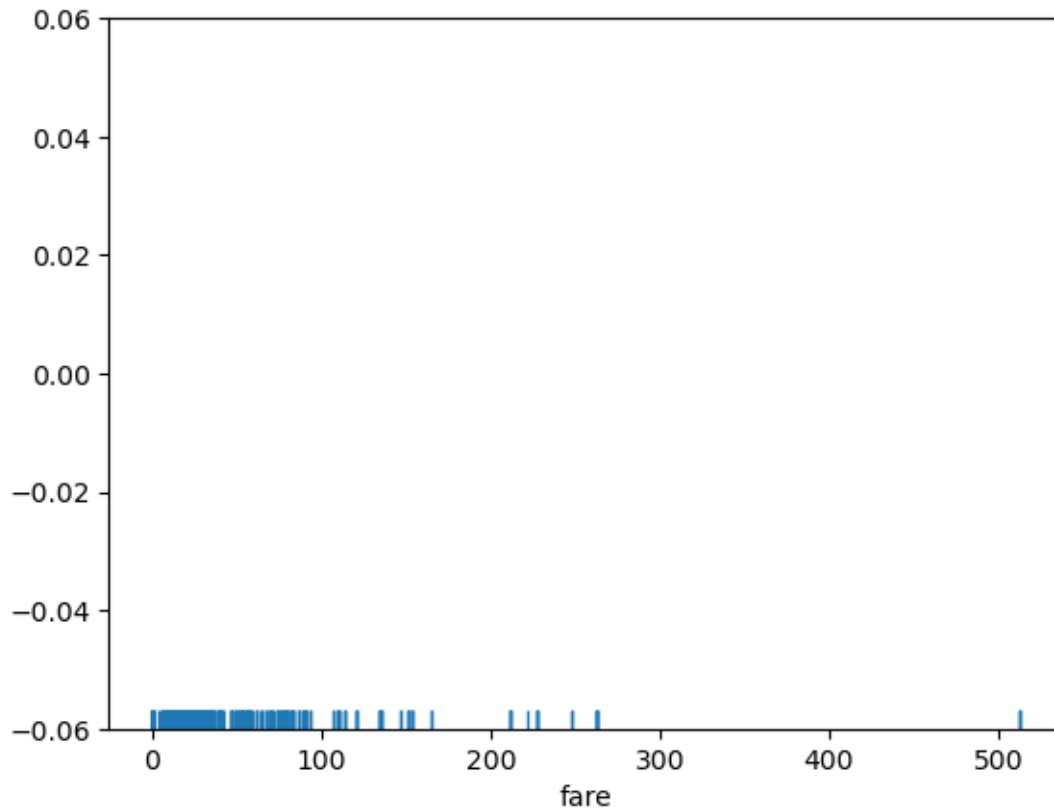


#### 4. Rog Plut (Pour voir la densite et la distribution d'une variable quantitative) L'intrigue du tapis

Le rugplot() est utilisé pour dessiner de petites barres le long de l'axe des x pour chaque point de l'ensemble de données

```
[13]: sns.rugplot(df['fare'])
```

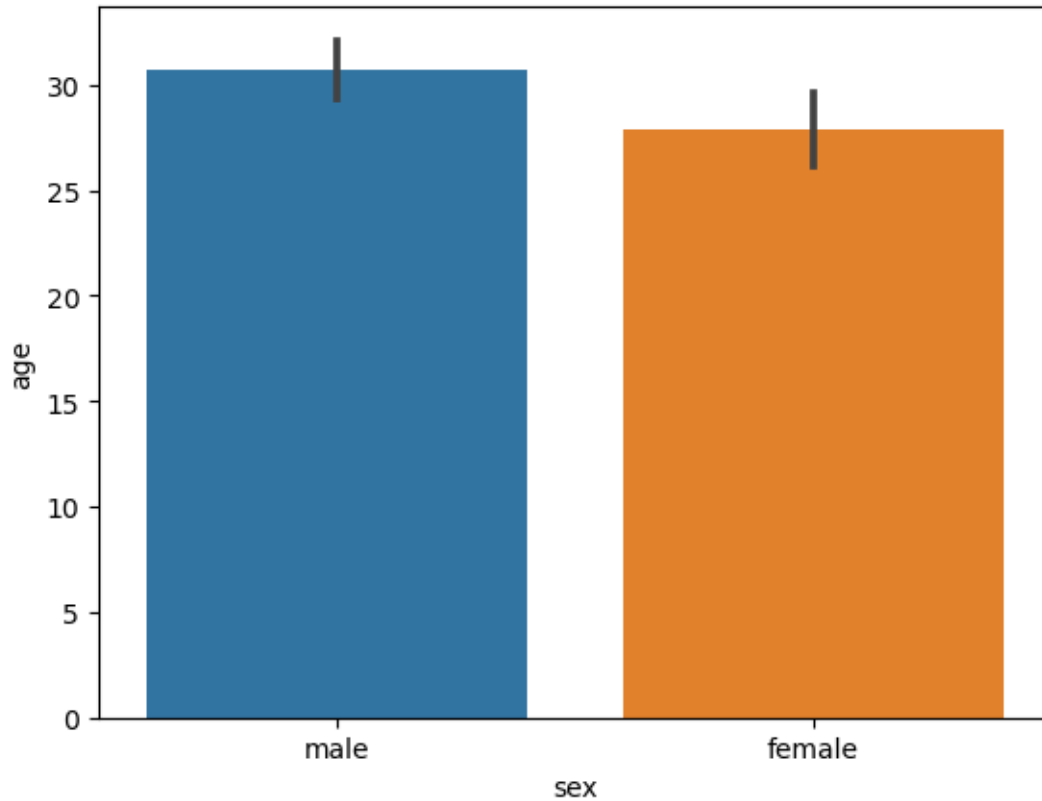
```
[13]: <Axes: xlabel='fare'>
```



**5. Bar plot pour voir la representation d'une variable quantitative sur base d'une variable qualitative** Le `barplot()` est utilisé pour afficher la valeur moyenne de chaque valeur dans une colonne catégorielle, par rapport à une colonne numérique. Le premier paramètre est la colonne catégorielle, le deuxième paramètre est la colonne numérique tandis que le troisième paramètre est l'ensemble de données

```
[14]: sns.barplot(x = 'sex', y = 'age', data = df) #affiche la moyenne d'age par sex
```

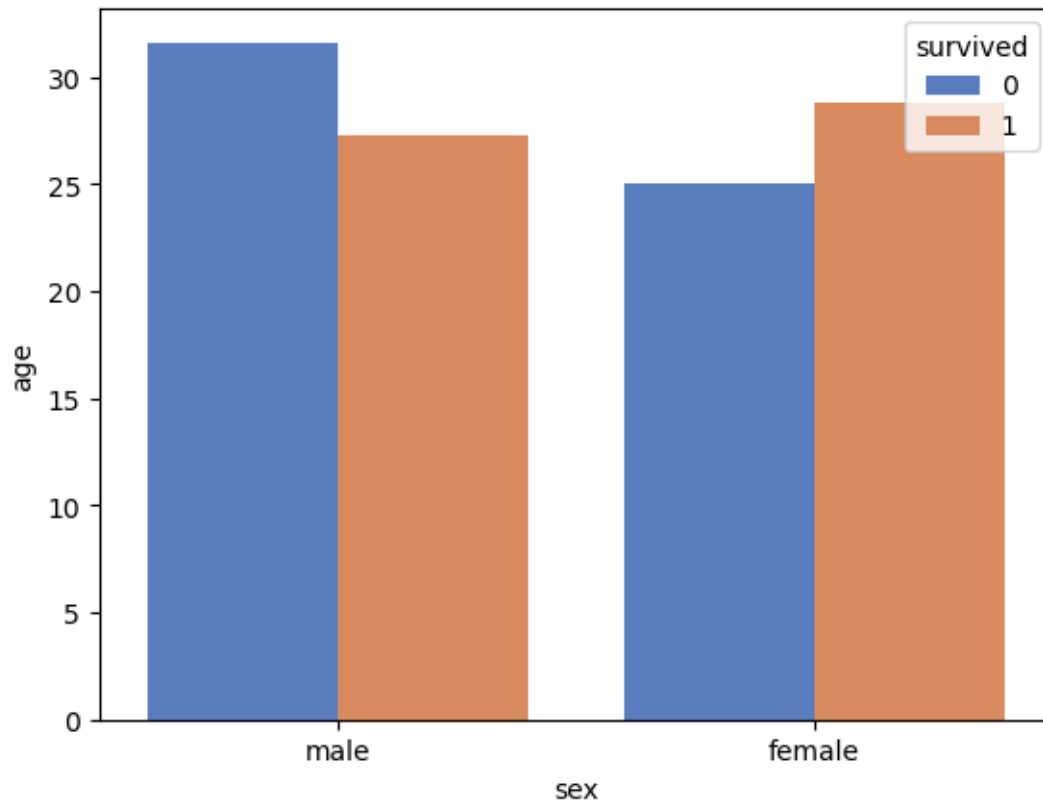
```
[14]: <Axes: xlabel='sex', ylabel='age'>
```



**Barplot combiné pour avoir une situation un peu plus détaillée** Les diagrammes à barres sont utilisés pour visualiser la relation entre une variable catégorielle et une variable continue. Dans un diagramme à barres, chaque barre représente la moyenne ou la médiane (ou toute agrégation) de la variable continue pour chaque catégorie. Dans Seaborn, les diagrammes à barres peuvent être créés à l'aide de la fonction `barplot()`.

```
[28]: sns.barplot(x = 'sex', y = 'age', hue = 'survived', ci=None, palette="muted",  
↳ data = df)
```

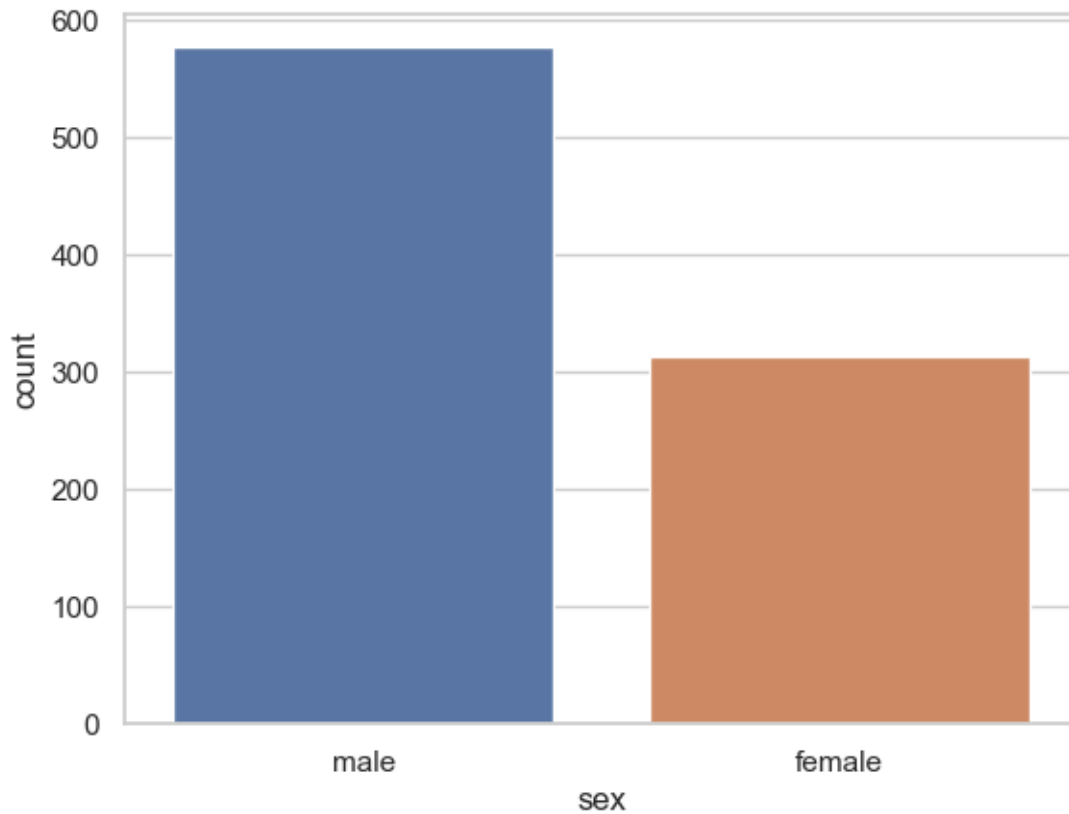
```
[28]: <Axes: xlabel='sex', ylabel='age'>
```



**6. Count Plot (Pour voir la situation d'une variable qualitative)** The count plot is similar to the bar plot, however it displays the count of the categories in a specific column

```
[78]: sns.set(style = "whitegrid")
      sns.countplot(x = 'sex', data = df)
```

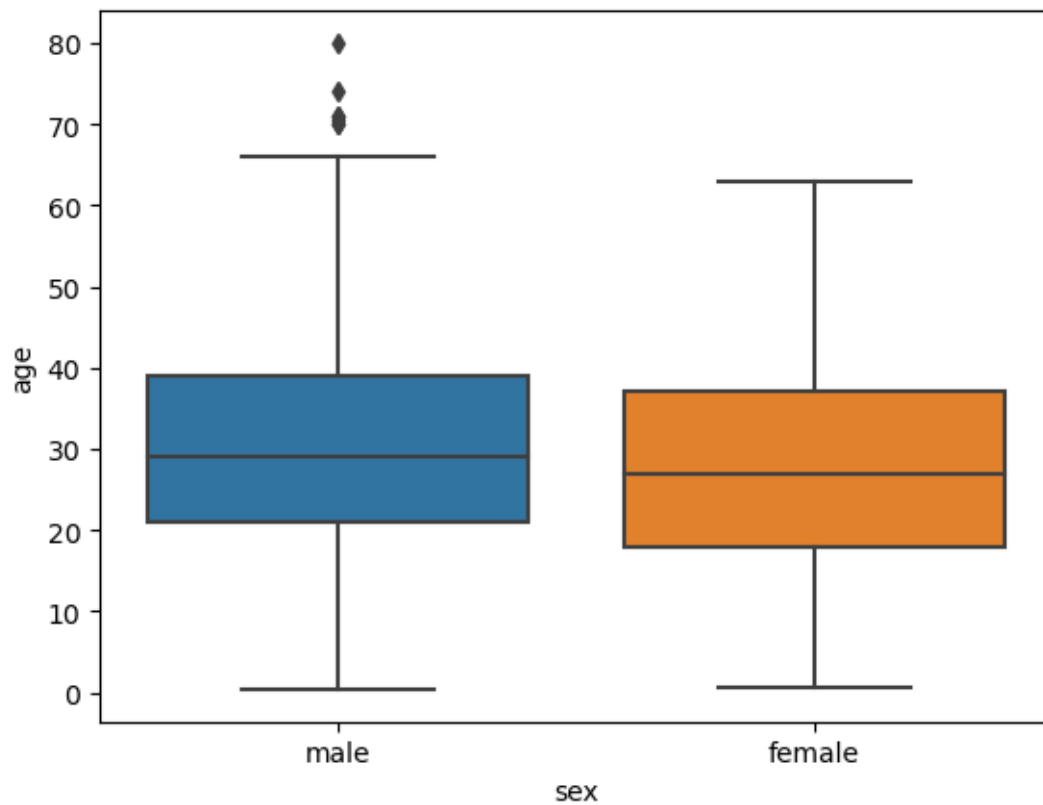
```
[78]: <Axes: xlabel='sex', ylabel='count'>
```



**7. BoxPlot()** Pour voir la distribution d'une variable qualitative par rapport a une variable quanti sous forme de quartile Les boîtes à moustaches sont un type de visualisation qui montre la distribution d'un ensemble de données. Ils sont couramment utilisés pour comparer la distribution d'une ou plusieurs variables dans différentes catégories.

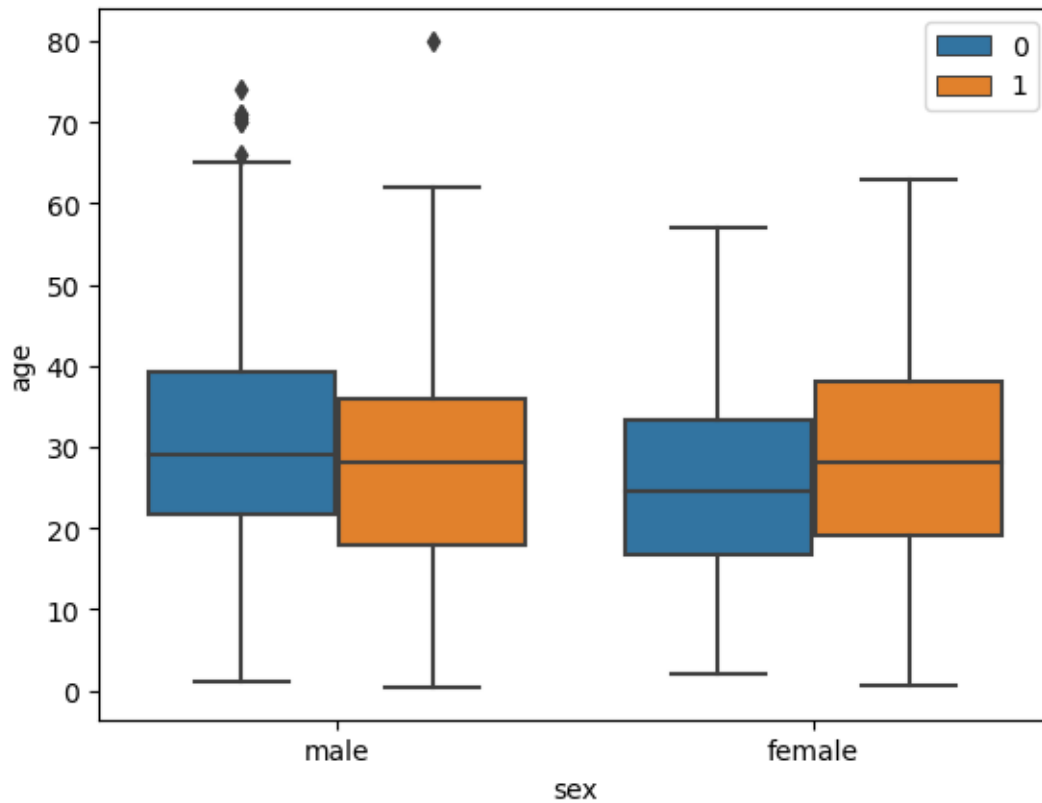
```
[16]: sns.boxplot(x = 'sex', y = 'age', data = df)
```

```
[16]: <Axes: xlabel='sex', ylabel='age'>
```



Ajouter une variable pour la categorisation de nos results dans les quartiles

```
[17]: sns.boxplot(x = 'sex', y = 'age', data = df, hue = 'survived')  
plt.legend()  
plt.show()
```



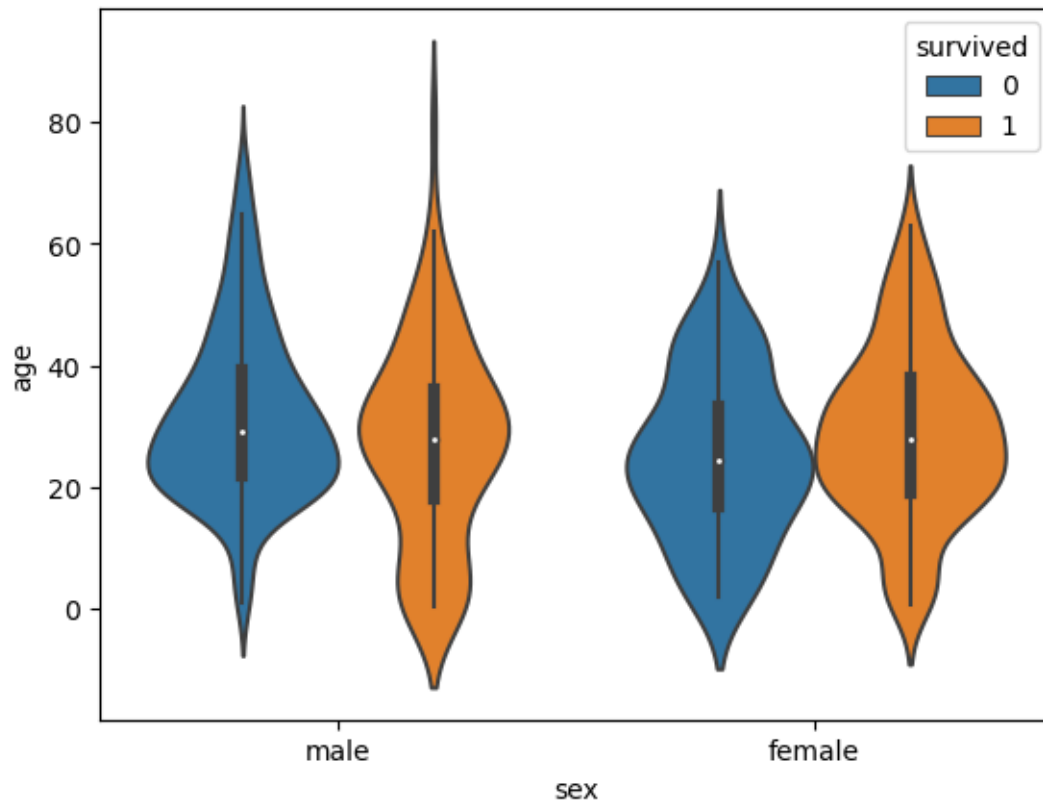
## 8. ViolinPlot (Idem que BoxPlot mais avec plus de details par rapport aux boxplot)

Un graphique en violon est un type de visualisation de données qui combine des aspects des boîtes à moustaches et des diagrammes de densité. Il affiche une estimation de la densité des données, généralement lissée par un estimateur de densité par noyau, ainsi que l'écart interquartile (IQR) et la médiane sous forme de boîte à moustaches.

La largeur du violon représente l'estimation de la densité, les parties plus larges indiquant une densité plus élevée, et l'IQR et la médiane sont représentés par un point blanc et une ligne à l'intérieur du violon.

```
[18]: sns.violinplot(x = 'sex', y = 'age', data = df, hue= 'survived')
```

```
[18]: <Axes: xlabel='sex', ylabel='age'>
```

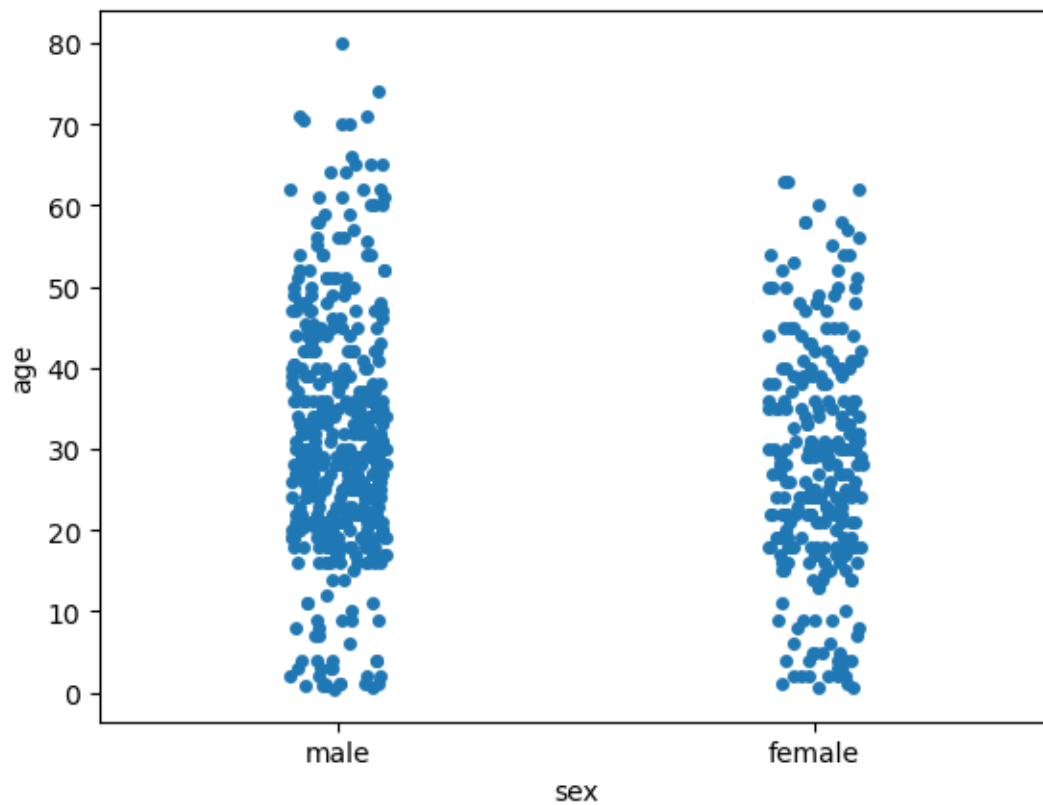


**9. Strip Plot pour voir la representation d'une variable quali sur base d'une variable quanti** le diagramme en bandes dessine un nuage de points où l'une des variables est catégorielle. Nous avons vu des nuages de points dans les sections du tracé conjoint et du tracé en paires où nous avons deux variables numériques. Le diagramme en bandes est différent dans le sens où l'une des variables est catégorielle dans ce cas, et pour chaque catégorie de la variable catégorielle, vous verrez un nuage de points par rapport à la colonne numérique.

```
[20]: sns.stripplot(x = 'sex', y = 'age', data = df, jitter = True)
```

```
[20]: <Axes: xlabel='sex', ylabel='age'>
```

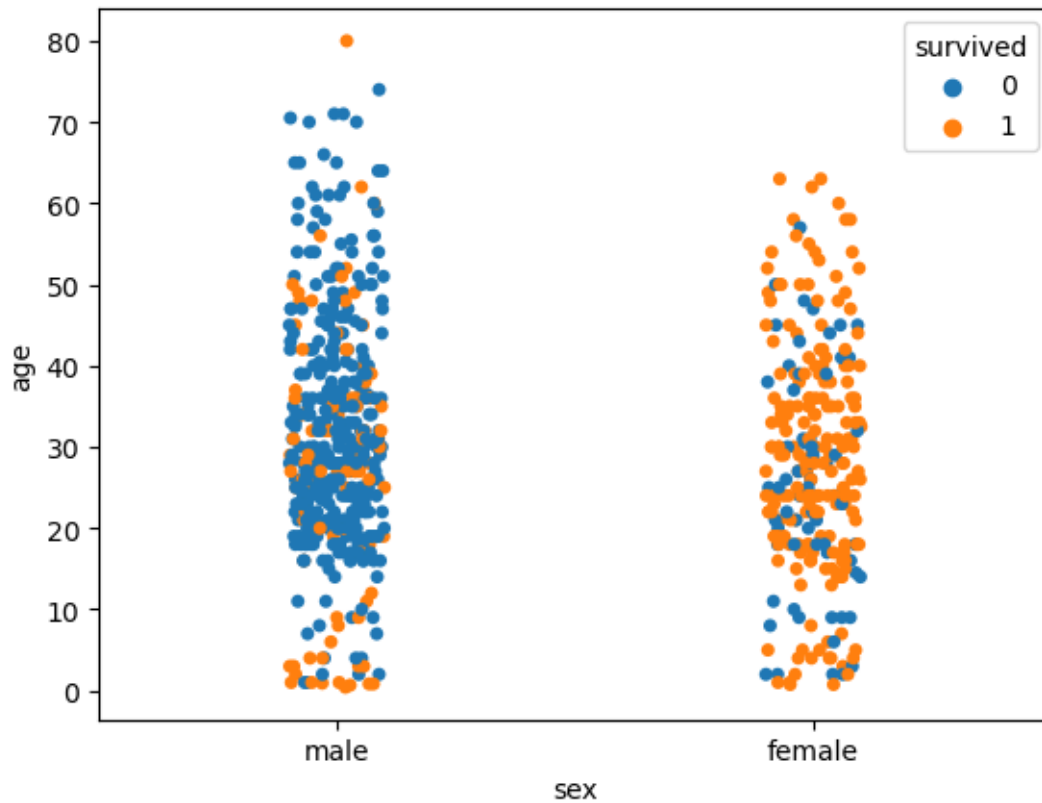




Ajouter une categorie

```
[23]: sns.stripplot(x = 'sex', y = 'age', data = df, hue = 'survived')
```

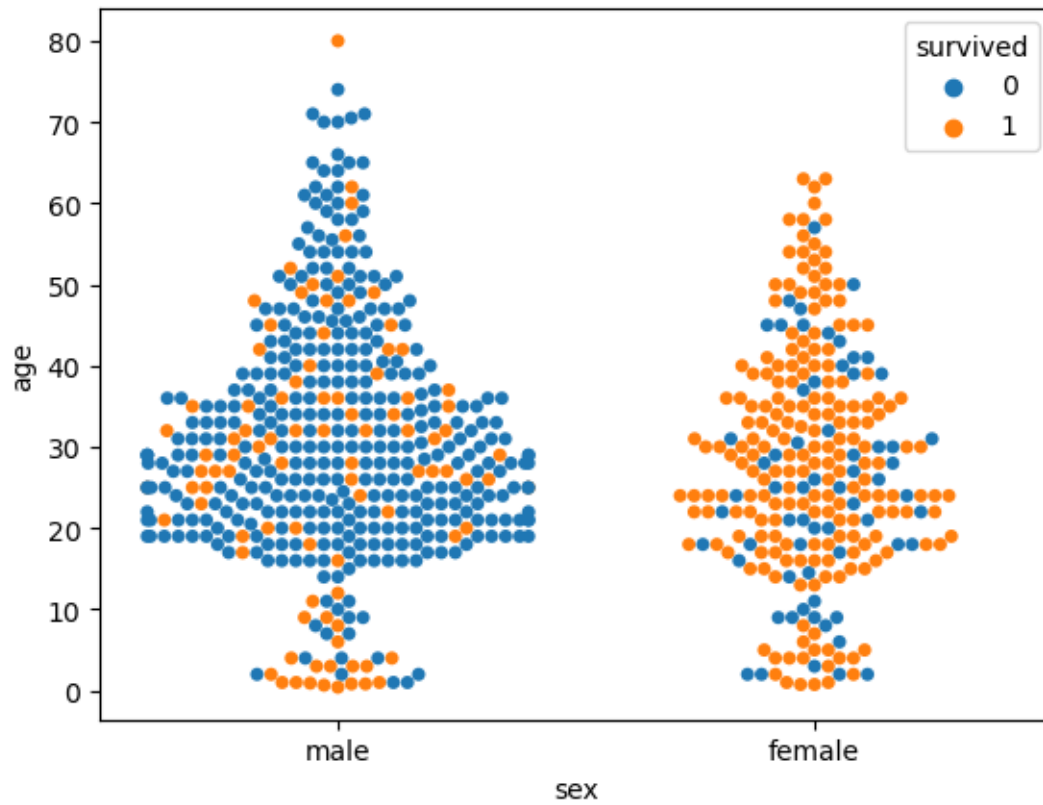
```
[23]: <Axes: xlabel='sex', ylabel='age'>
```



**10. The Swarm Plot (Pour la distribution et le mm type d'étude que violin, strip et box plot)** L'intrigue en essaim est une combinaison des intrigues en bande et en violon. Dans les parcelles d'essaim, les points sont ajustés de manière à ne pas se chevaucher

```
[25]: sns.swarmplot(x = 'sex', y = 'age', data = df, hue = 'survived')
```

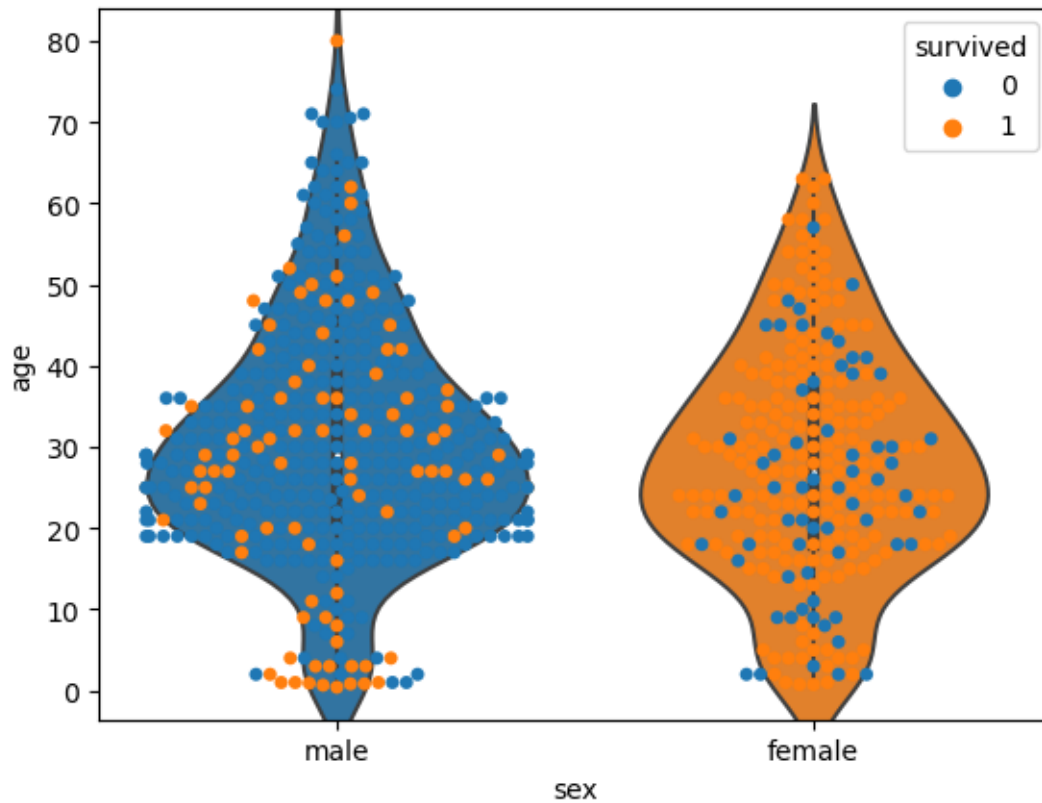
```
[25]: <Axes: xlabel='sex', ylabel='age'>
```



### Combiner Swarm et Violin Plot

```
[26]: sns.violinplot(x = 'sex', y = 'age', data = df)
      sns.swarmplot(x = 'sex', y = 'age', data = df, hue = 'survived')
```

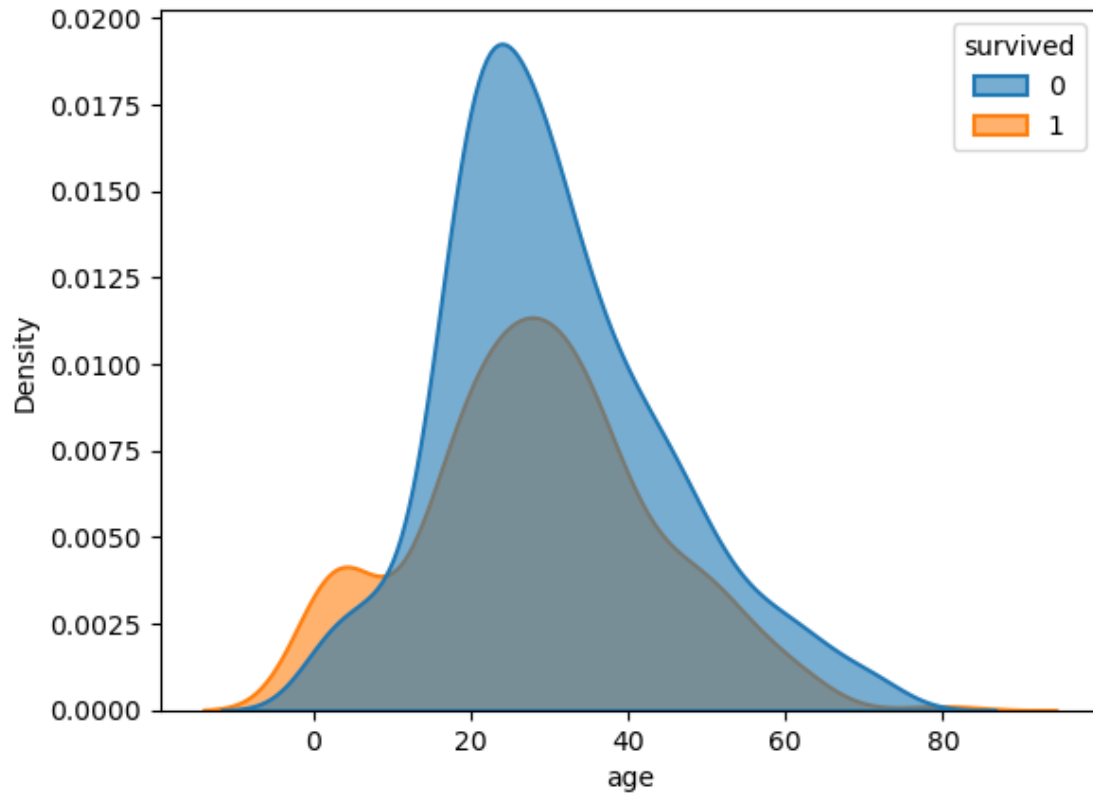
```
[26]: <Axes: xlabel='sex', ylabel='age'>
```



**11. kdeplot pour voir la densité de distribution d'une variable améliorée** Les diagrammes de densité, également connus sous le nom de diagrammes de densité de noyau, sont un type de visualisation de données qui affiche la distribution d'une variable continue. Ils sont similaires aux histogrammes, mais au lieu de représenter les données sous forme de barres, les diagrammes de densité utilisent une courbe lisse pour estimer la densité des données. Dans Seaborn, des diagrammes de densité peuvent être créés à l'aide de la fonction `kdeplot()`.

```
[27]: sns.kdeplot(data = df, x = 'age', hue = 'survived', fill=True, alpha=0.6,
↳ linewidth=1.5)
```

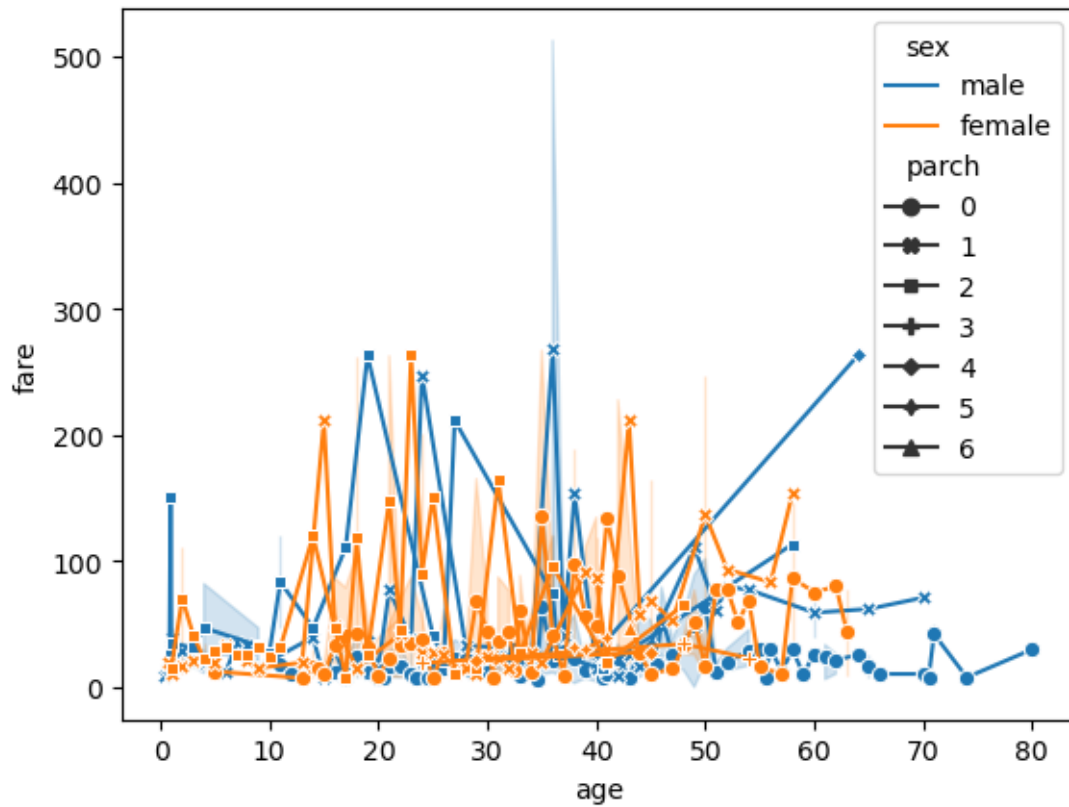
```
[27]: <Axes: xlabel='age', ylabel='Density'>
```



**12.Lineplot pour la visualisation des tendances ou l'evolution des donnees sur base du temps** Les tracés linéaires sont utilisés pour visualiser les tendances des données dans le temps ou d'autres variables continues. Dans un tracé linéaire, chaque point de données est relié par une ligne, créant une courbe lisse. Dans Seaborn, les tracés linéaires peuvent être créés à l'aide de la fonction `lineplot()`.

```
[30]: sns.lineplot(x = 'age', y = 'fare', hue = 'sex', style = 'parch', markers=True,
↳dashes=False, data=df)
```

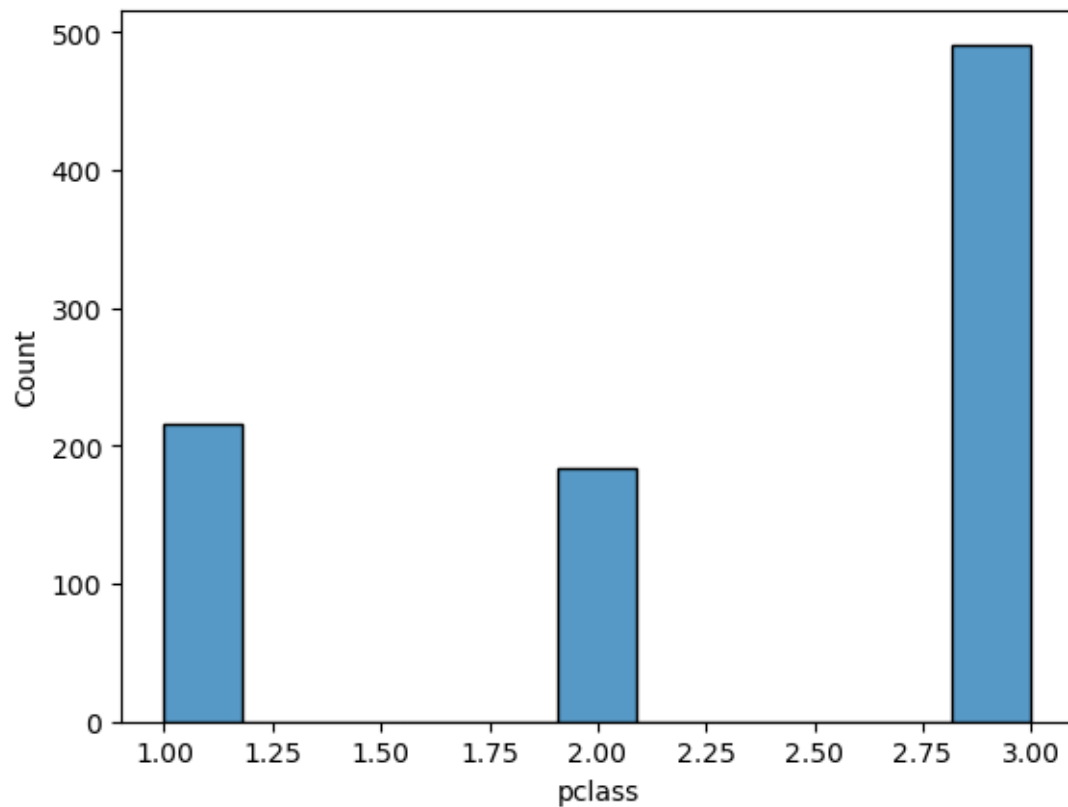
```
[30]: <Axes: xlabel='age', ylabel='fare'>
```



### 13. Histplot pour voir la distribution des variables quantitatives continues

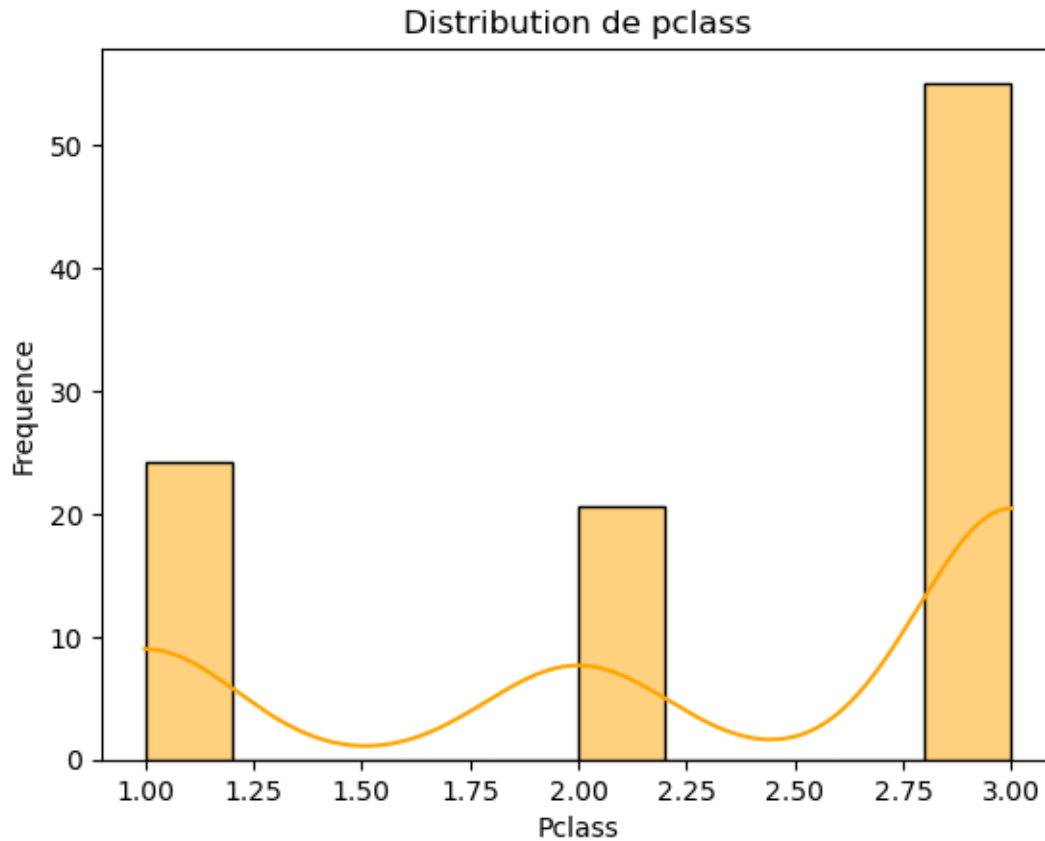
```
[31]: sns.histplot(x = "pclass", data = df)
```

```
[31]: <Axes: xlabel='pclass', ylabel='Count'>
```



Personnaliser le histplot pour voir la ligne tendance de la distribution avec l'option `stat percent` pour voir la proportion en pourcentage

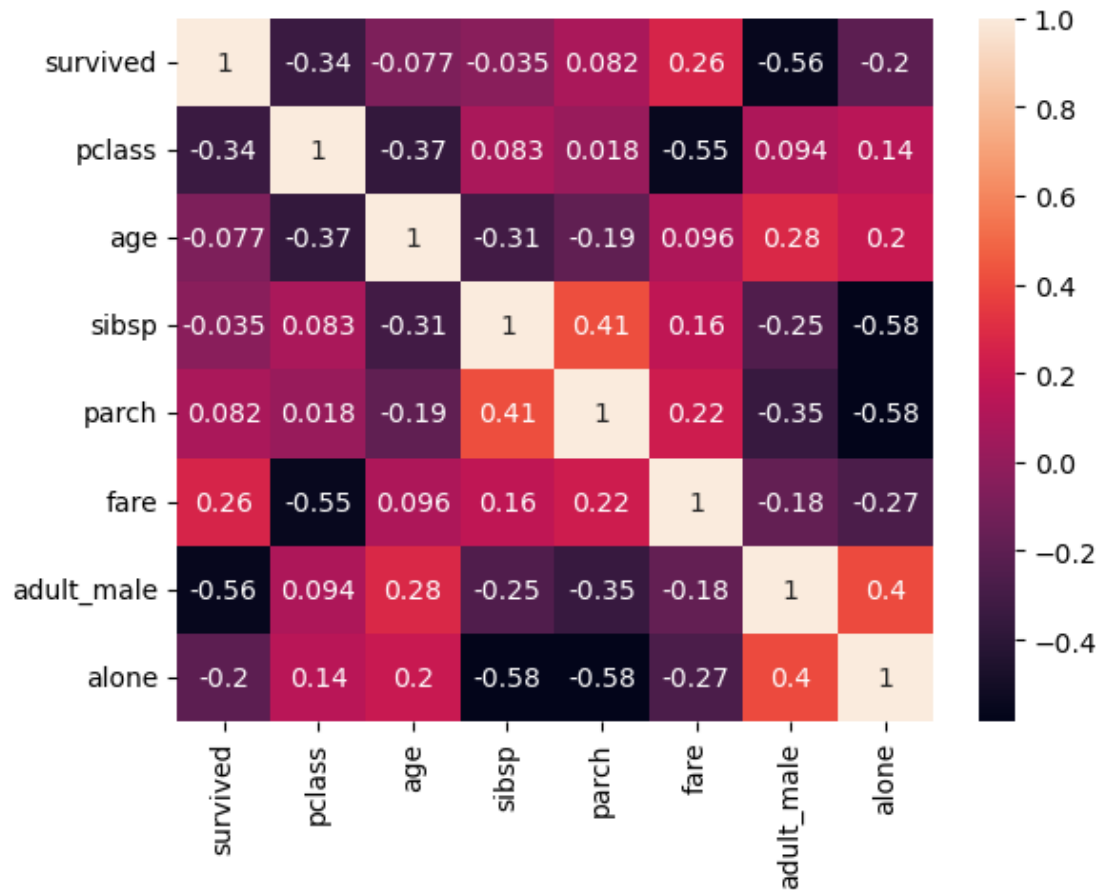
```
[67]: sns.histplot(x = "pclass", data = df, kde = True, color = "orange", bins = 10, stat = "percent")
plt.xlabel("Pclass")
plt.ylabel("Frequence")
plt.title("Distribution de pclass")
plt.show()
```



**14. Heatmap pour la visualisation de la corrélation entre les variables** Une carte thermique est une représentation graphique de données qui utilise des couleurs pour représenter la valeur d'une variable dans un espace bidimensionnel. Les cartes thermiques sont couramment utilisées pour visualiser la corrélation entre différentes variables dans un ensemble de données.

```
[66]: #Avoir la corrélation entre toutes les variables du dataset  
corr = df.corr()  
  
sns.heatmap(corr, annot = True)  
  
plt.show()
```



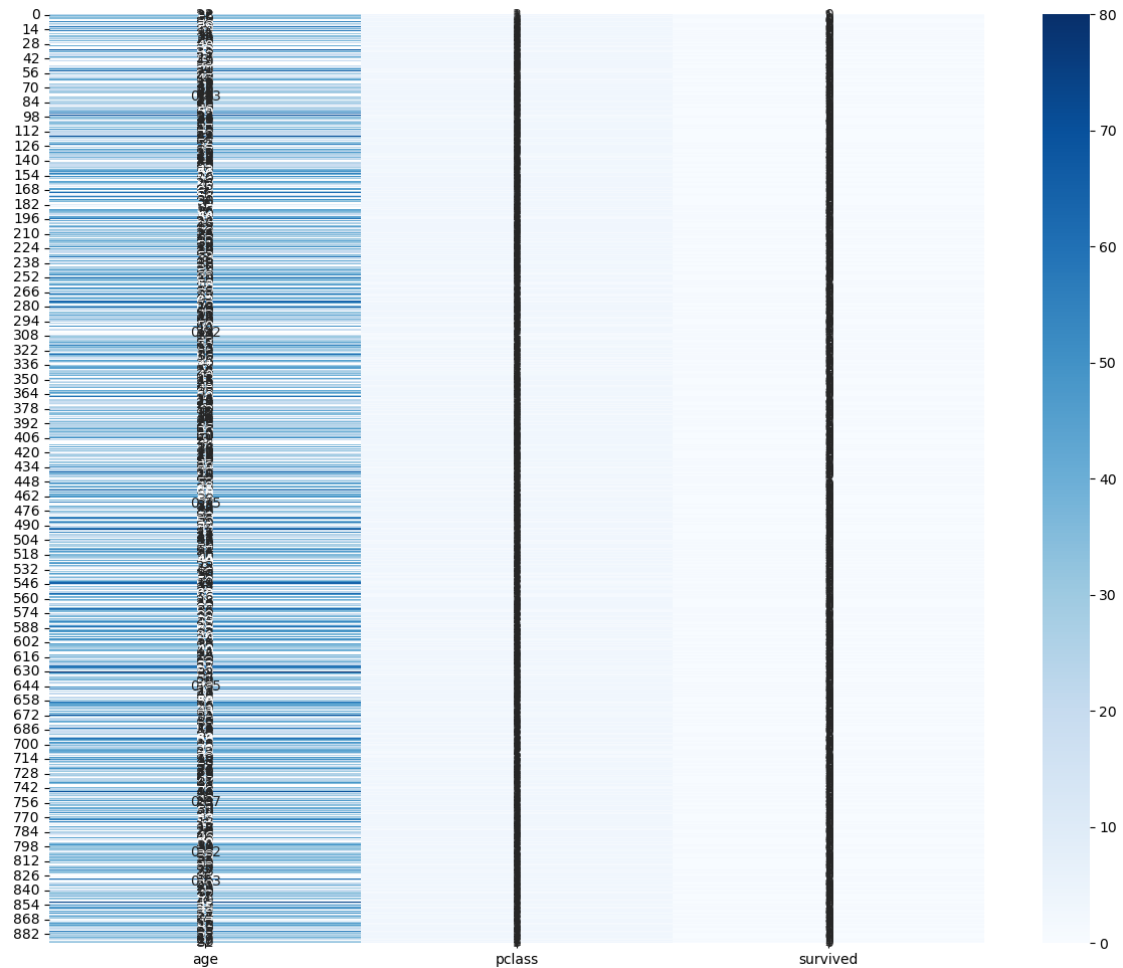


```
[44]: #Pivoter les donnees et maintenir seulement les datas pour lesquelles on veut
      ↪ faire la correlation

      #df_to_correlate = df.pivot('age', 'pclass', 'survived')
      plt.figure(figsize = (15, 12))
      df_to_correlaet = df[["age", "pclass", "survived"]]

      sns.heatmap(df_to_correlaet, cmap='Blues', annot =True)

      plt.show()
```



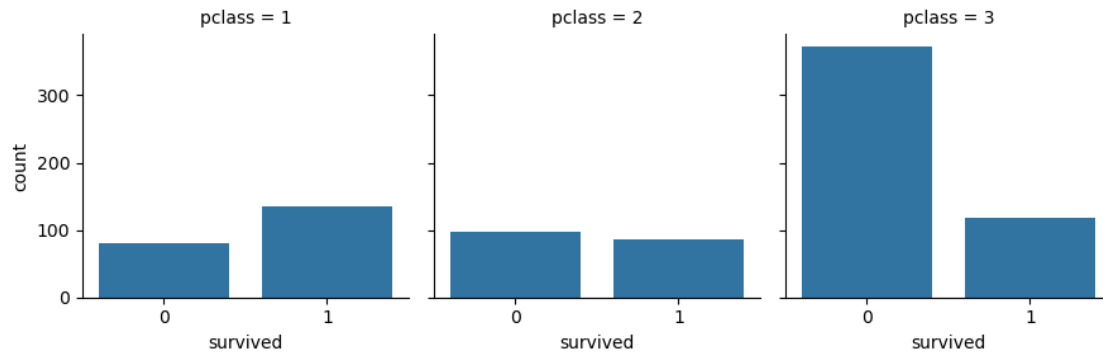
**15. Facetgrid pour la visulation de la distribution d'une variable ainsi que la relation entre deux variables avec un level des variables qualitatives** FacetGrid est un puissant outil marin qui vous permet de visualiser la distribution d'une variable ainsi que la relation entre deux variables, à travers des niveaux de variables catégorielles supplémentaires.

FacetGrid crée une grille de sous-tracés basée sur les valeurs uniques de la variable catégorielle spécifiée.

```
[57]: g = sns.FacetGrid(df, col ="pclass")

      g.map(sns.countplot, "survived")
```

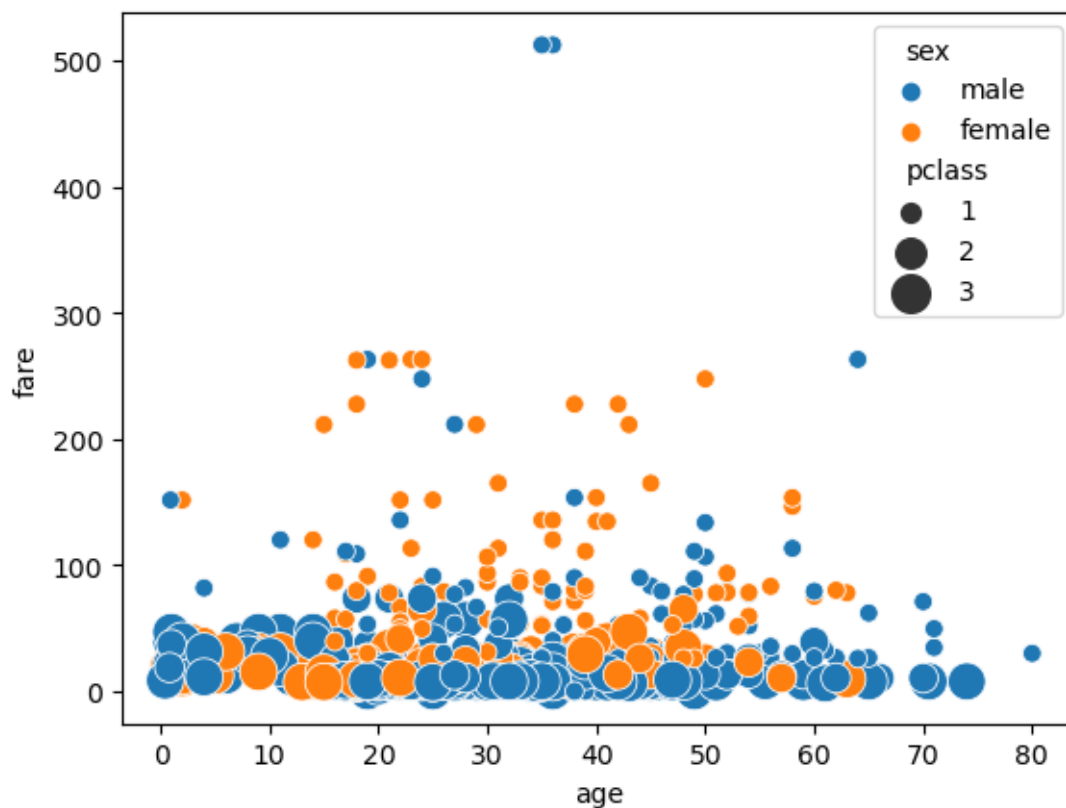
```
[57]: <seaborn.axisgrid.FacetGrid at 0x1cfc5478280>
```



**16. Scatterplot pour voir la relation entre deux variables quantitatives** Les nuages de points sont utilisés pour visualiser la relation entre deux variables continues. Chaque point du graphique représente un seul point de données, et la position du point sur les axes x et y représente les valeurs des deux variables.

```
[56]: sns.scatterplot(x = "age", y = "fare", data = df, hue="sex", size = "pclass",
    ↪ sizes = (50,200))
```

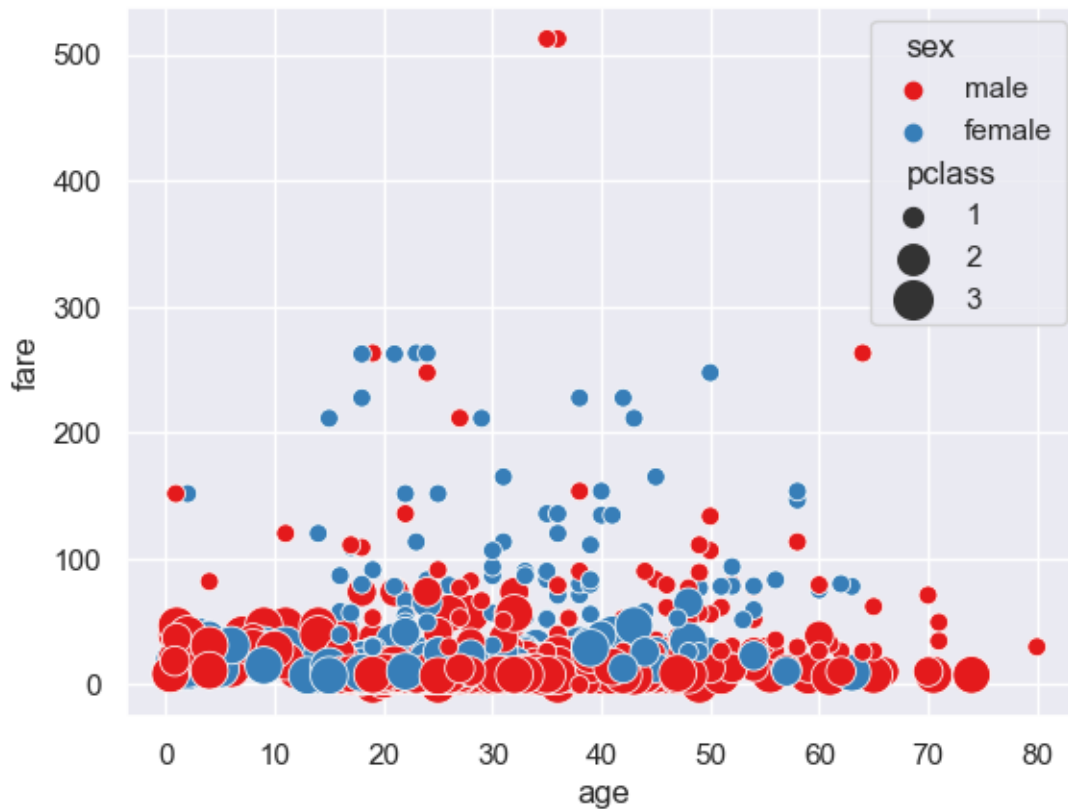
```
[56]: <Axes: xlabel='age', ylabel='fare'>
```



Personnaliser le scatter plot , cahngement de la palette (mettre a Set2 ou 3 ou 1), ajuster le size de la figure et les Annotations, sns.set(“dark”) pour parametrer le theme

```
[75]: sns.set(style="darkgrid")
sns.scatterplot(x = "age", y = "fare", data = df, hue="sex", size = "pclass",
               sizes = (50,200), palette = "Set1")
```

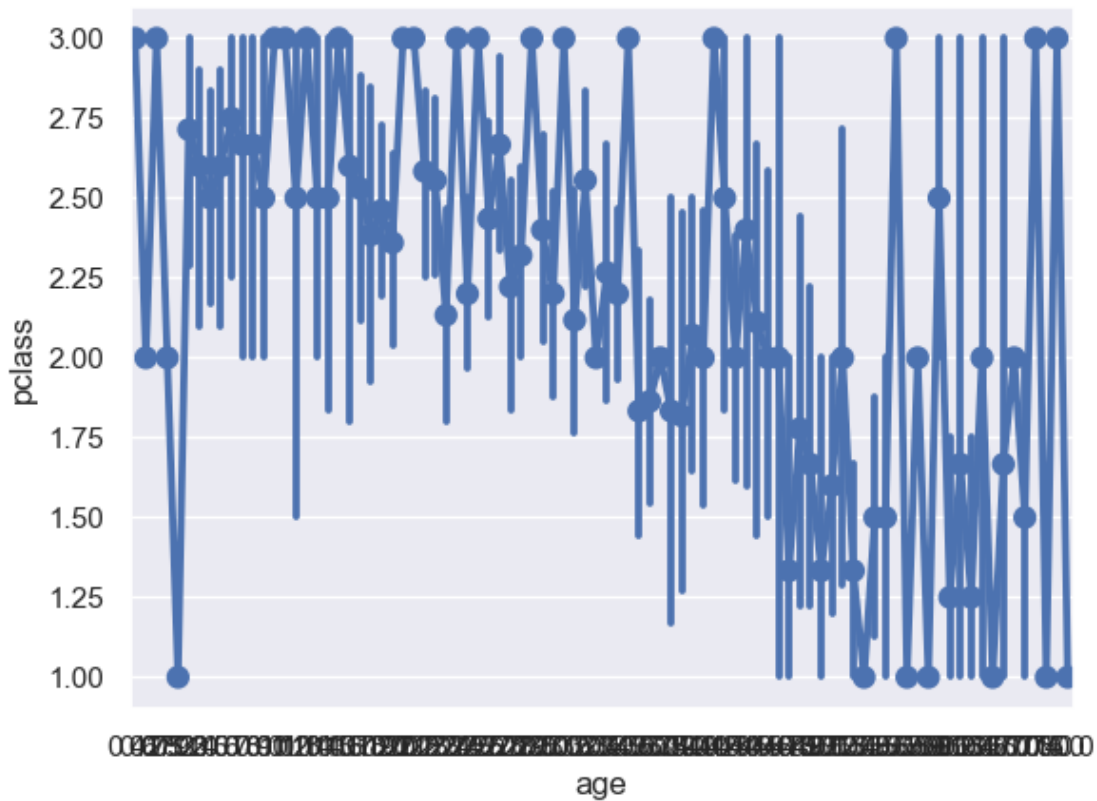
```
[75]: <Axes: xlabel='age', ylabel='fare'>
```



17. Pointplot pour voir la correlation entre deux variables et montrer l’incertitude

```
[77]: sns.pointplot(x = "age", y = "pclass", data = df)
```

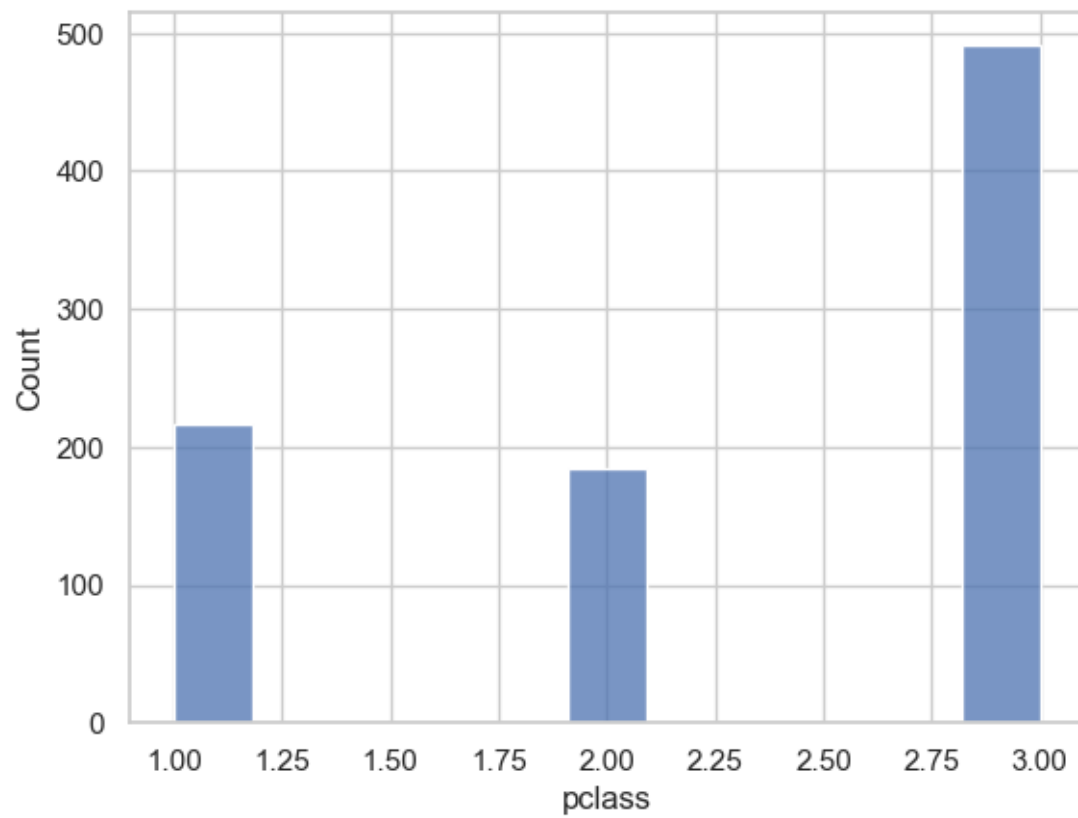
```
[77]: <Axes: xlabel='age', ylabel='pclass'>
```



**18. Hist plot()** Les histogrammes visualisent la distribution d'une variable continue. Dans un histogramme, les données sont divisées en groupes et la hauteur de chaque groupe représente la fréquence ou le nombre de points de données dans ce groupe. Dans Seaborn, les histogrammes peuvent être créés à l'aide de la fonction `histplot()`

```
[80]: sns.histplot(x="pclass", data=df)
```

```
[80]: <Axes: xlabel='pclass', ylabel='Count'>
```



[ ]: