

Machine Learning II

Introduction à matplotlib

Nicolas Bourgeois

Télécharger

Data and Cheatsheets :

`ouralou.fr/Resources/epita/C2.zip`

Exercice

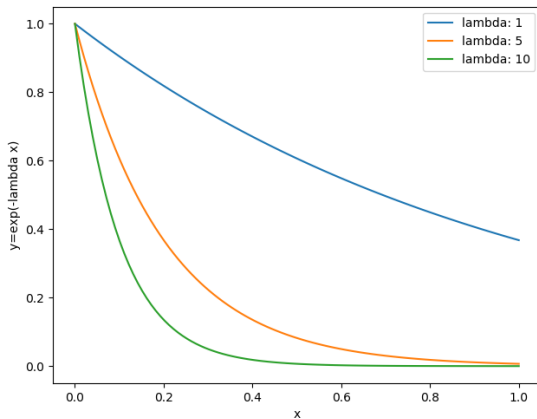
Exercice

Affichez les fonctions $x \mapsto e^{-\lambda x}$ pour diverses valeurs du paramètre λ . N'oubliez pas l'axe et la légende.

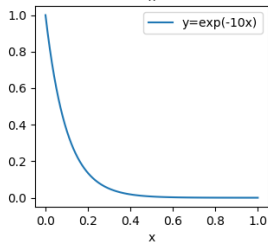
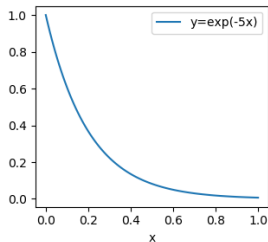
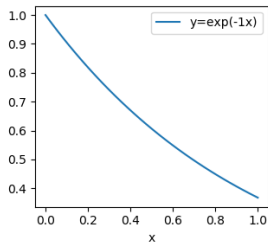
Exercice

Même question, mais les courbes doivent être sur des sous-figures différentes et non superposées.

Résultat attendu (1)



Résultat attendu (2)



Solution - Première partie

```
from matplotlib import pyplot as plt
import numpy as np
from math import exp
plt.ylabel("y=exp(-lambda_x)")
plt.xlabel("x")
x=np.linspace(0,1,100)
for k in [1,5,10]:
    plt.plot(x,np.vectorize(lambda z:exp(-k*z))(x),
             label="lambda:_{0}".format(k))
plt.legend()
plt.show()
```

Solution -Deuxième partie

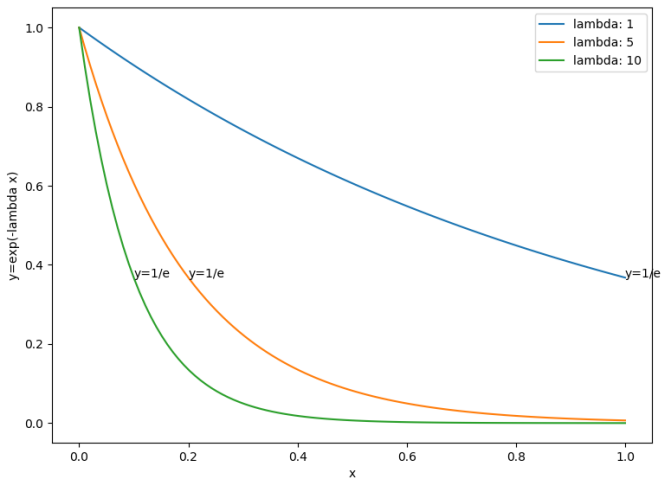
```
from matplotlib import pyplot as plt
import numpy as np
from math import exp
fig = plt.figure()
x=np.linspace(0,1,100)
for i,k in enumerate([1,5,10]):
    fig.add_subplot(2,2,i+1)
    plt.plot(x,np.vectorize(lambda z:exp(-k*z))(x),
        label="y=exp(-{0}x)".format(k))
    plt.xlabel("x")
    plt.legend()
plt.show()
```

Exercice

Exercice

Reprenez les courbes du premier exercice et annotez-les pour distinguer les points $y = e^{-1}$.

Résultat attendu



Solution

```
from matplotlib import pyplot as plt
import numpy as np
from math import exp
plt.ylabel("y=exp(-lambda_x)")
plt.xlabel("x")
x=np.linspace(0,1,100)
for k in [1,5,10]:
    plt.plot(x,np.vectorize(lambda z:exp(-k*z))(x),
             label="lambda: {0}".format(k))
    plt.annotate("y=1/e", (1/k, exp(-1)))
plt.legend()
plt.show()
```

Exercice

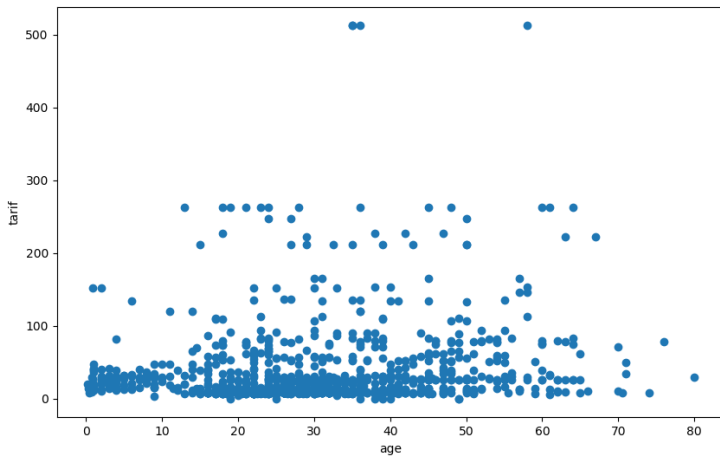
Exercice

Importez les données de data1.csv dans un dataframe et affichez leur dispersion selon les axes age et prix du billet.

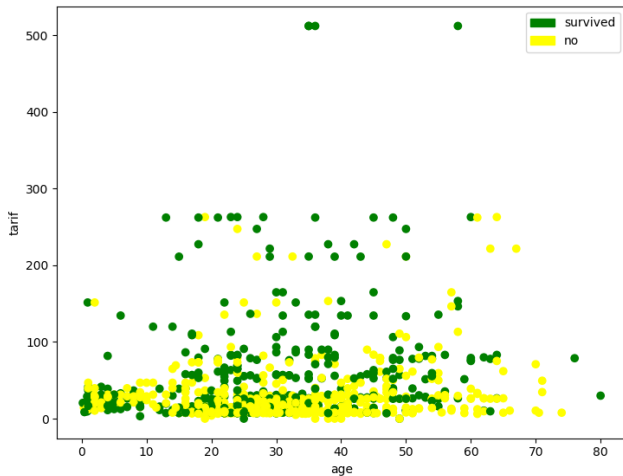
Exercice

Modifiez la couleur et ajoutez une légende de façon à faire apparaître l'information si la personne a survécu ou non.

Résultat attendu (1)



Résultat attendu (2)



Solution - première partie

```
import pandas as pd
from matplotlib import pyplot as plt
df1 = pd.read_csv('./C2/data1.csv')
plt.scatter(df1.age, df1.fare)
plt.ylabel("tarif")
plt.xlabel("age")
plt.show()
```

Solution - deuxième partie

```
import pandas as pd
from matplotlib import pyplot as plt
from matplotlib import patches
df1 = pd.read_csv('./C2/data1.csv')
def colors(d):
    return {1: 'green', 0: 'yellow'}[d]
plt.scatter(df1.age, df1.fare,
            c=df1.survived.apply(colors))
plt.ylabel("tarif")
plt.xlabel("age")
colnames = [patches.Patch(color='green',
                           label='survived'),
            patches.Patch(color='yellow', label='no')]
plt.legend(handles=colnames)
plt.show()
```

Exercice

Exercice

Utilisez une scatter matrix pour croiser les données age et prix du billet, tout en gardant la couleur pour évaluer la survie.

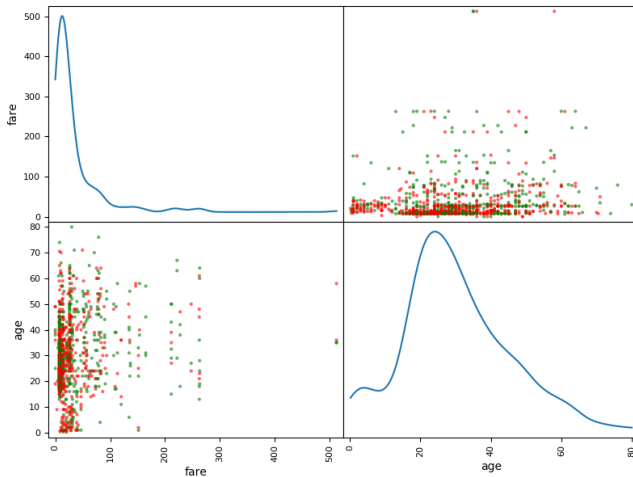
Exercice

Ajoutez le genre pour rendre la scatter matrix utile. Vous devrez convertir le type de données.

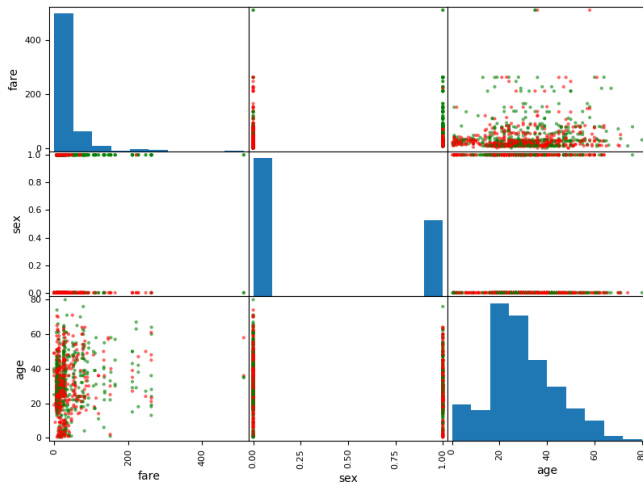
Exercice

Ajoutez un facteur de dispersion aléatoire des points autour de la valeur genre de façon à mieux les visualiser.

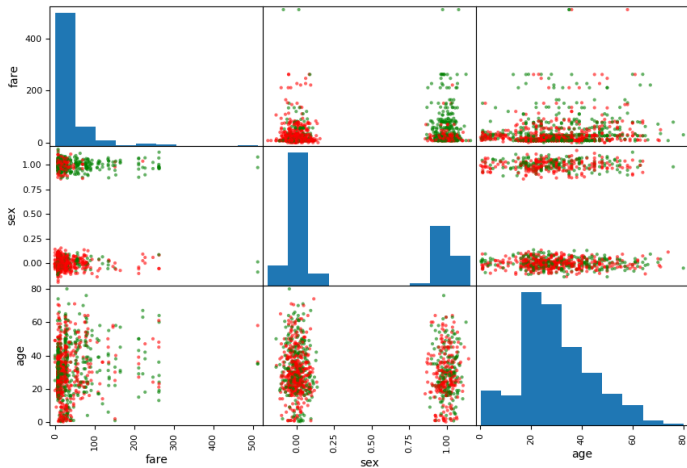
Résultat attendu (1)



Résultat attendu (2)



Résultat attendu (3)



Solution - première partie

```
import pandas as pd
from matplotlib import pyplot as plt
from pandas.plotting import scatter_matrix
titanic = pd.read_csv(' ./C2/data1.csv')[['fare', 'age', 'survived']]
colMap={0:"red",1:"green"}
colors=list(map(lambda x:colMap.get(x),titanic.survived))
scatter_matrix(titanic[['fare', 'age']], alpha=0.6,
               c=colors, diagonal='kde')
plt.show()
```

Solution - deuxième partie

```
import pandas as pd
from matplotlib import pyplot as plt
from pandas.plotting import scatter_matrix
titanic = pd.read_csv('./C2/data1.csv')
colMap={0:"red",1:"green"}
colors=list(map(lambda x:colMap.get(x),titanic.survived))
titanic.sex = titanic.sex.apply(
    lambda x:{'male':0,'female':1}[x])
scatter_matrix(titanic[['fare','sex','age']],
    alpha=0.6,c=colors,diagonal='hist')
plt.show()
```

Solution - troisième partie

```

import pandas as pd
from matplotlib import pyplot as plt
from pandas.plotting import scatter_matrix
from random import gauss
titanic = pd.read_csv( './C2/data1.csv' )
colMap={0:"yellow",1:"green"}
def colors(d):
    return {1:'green',0:'yellow'}[d]
titanic.sex = titanic.sex.apply(
    lambda x:{ 'male':0, 'female':1}[x])
titanic['sex'] = titanic['sex'].apply(
    lambda x:x+gauss(0,0.05))
titanic['survived'] = titanic['survived'].apply(colors)
plt.scatter(titanic['age'],titanic['fare'],c=titanic.surviv
for x in titanic.index:
    if titanic.fare[x] > 400:
        print(titanic.survived[x])
plt.show()
```