

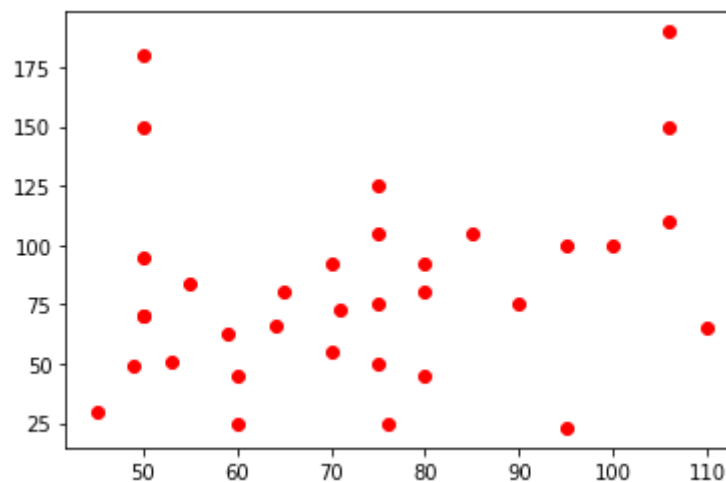
1. Pokemon

```
In [2]: import matplotlib.pyplot as plt
import csv

fichier = open('pokemons.csv', 'r', encoding = 'UTF-8')
t = csv.DictReader(fichier, delimiter=';')
pokemons = [dict(ligne) for ligne in t] # création et
construction du tableau par compréhension
fichier.close()

pokemons_eau = [p for p in pokemons if p['Type'] == 'Eau']
pokemons_psy = [p for p in pokemons if p['Type'] == 'Psy']

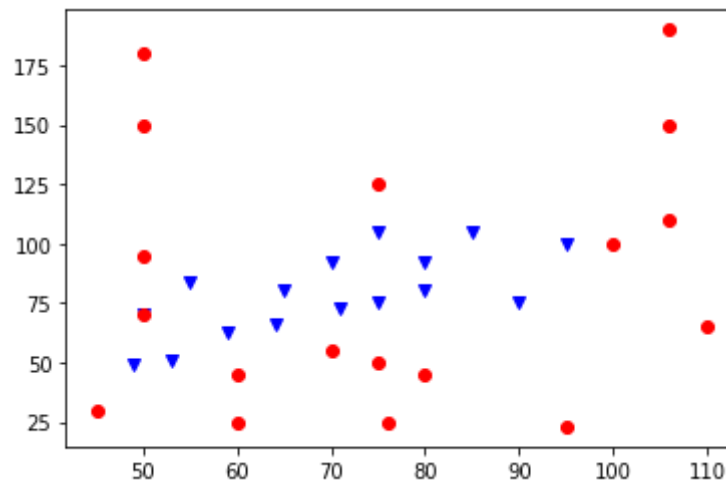
points_de_vie = [int(p['Points de vie']) for p in pokemons]
# construction du tableau des points de vie
valeur_attaque = [int(p['Attaque']) for p in pokemons] #
construction du tableau des valeurs d'attaque
plt.plot(points_de_vie, valeur_attaque, 'ro') #
construction du graphique
plt.show() # affichage du graphique
plt.close()
```



```
In [3]: import matplotlib.pyplot as plt
points_de_vie1 = [int(p['Points de vie']) for p in
pokemons_eau] # construction du tableau des points de vie
valeur_attaque1 = [int(p['Attaque']) for p in pokemons_eau]
# construction du tableau des valeurs d'attaque
points_de_vie2 = [int(p['Points de vie']) for p in
pokemons_psy] # construction du tableau des points de vie
```

```
valeur_attaque2 = [int(p['Attaque']) for p in pokemons_psy]
# construction du tableau des valeurs d'attaque

plt.plot(points_de_vie1, valeur_attaque1, 'vb') #
# construction du graphique
plt.plot(points_de_vie2, valeur_attaque2, 'ro') #
# construction du graphique
plt.show() # affichage du graphique
plt.close()
```



In [4]: `import matplotlib.pyplot as plt`

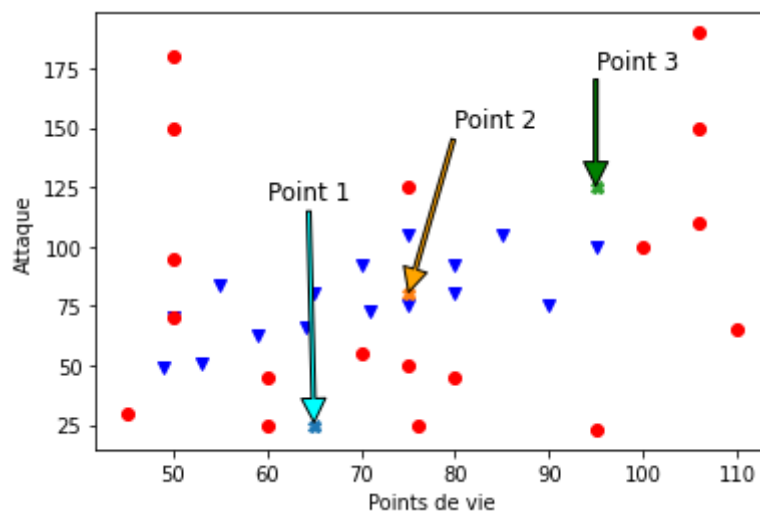
```
points_de_vie1 = [int(p['Points de vie']) for p in
pokemons_eau] # construction du tableau des points de vie
valeur_attaque1 = [int(p['Attaque']) for p in pokemons_eau]
# construction du tableau des valeurs d'attaque
points_de_vie2 = [int(p['Points de vie']) for p in
pokemons_psy] # construction du tableau des points de vie
valeur_attaque2 = [int(p['Attaque']) for p in pokemons_psy]
# construction du tableau des valeurs d'attaque

plt.plot(points_de_vie1, valeur_attaque1, 'vb') #
# construction du graphique
plt.plot(points_de_vie2, valeur_attaque2, 'ro') #
# construction du graphique
plt.plot(65,25, 'X') # construction du graphique
plt.plot(75,80, 'X') # construction du graphique
plt.plot(95,125, 'X') # construction du graphique
plt.xlabel('Points de vie')
plt.ylabel('Attaque')

plt.annotate('Point 1', xy=(65, 25), xytext=(60, 120),
fontSize=12,
            arrowprops={'facecolor':'cyan',
'width':2,'shrink':0.01})
```

```
plt.annotate('Point 2', xy=(75, 80), xytext=(80, 150),
            fontsize=12,
            arrowprops={'facecolor':'orange',
                        'width':2,'shrink':0.01})

plt.annotate('Point 3', xy=(95, 125), xytext=(95, 175),
            fontsize=12,
            arrowprops={'facecolor':'green',
                        'width':2,'shrink':0.01})
plt.show() # affichage du graphique
plt.close()
```



In [5]: `import matplotlib.pyplot as plt`

```
points_de_vie1 = [int(p['Points de vie']) for p in
pokemons_eau] # construction du tableau des points de vie
valeur_attaque1 = [int(p['Attaque']) for p in pokemons_eau]
# construction du tableau des valeurs d'attaque
points_de_vie2 = [int(p['Points de vie']) for p in
pokemons_psy] # construction du tableau des points de vie
valeur_attaque2 = [int(p['Attaque']) for p in pokemons_psy]
# construction du tableau des valeurs d'attaque

plt.plot(points_de_vie1, valeur_attaque1, 'vb') #
construction du graphique
plt.plot(points_de_vie2, valeur_attaque2, 'ro') #
construction du graphique

plt.plot(95,125, 'X') # construction du graphique
plt.xlabel('Points de vie')
plt.ylabel('Attaque')

plt.annotate('Nouvel élément à classifier', xy=(95, 125),
```

```

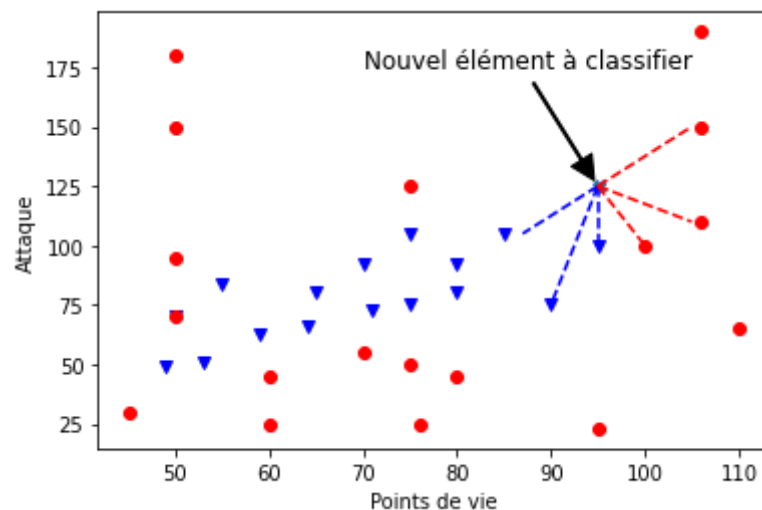
xytext=(70, 175), fontsize=12,
        arrowprops={'facecolor':'black',
                    'width':1, 'shrink':0.05})

plt.plot([95,95],[125,101],"b--")
plt.plot([95,90],[125,75],"b--")
plt.plot([95,87],[125,105],"b--")

plt.plot([95,100],[125,100],"r--")
plt.plot([95,105],[125,110],"r--")
plt.plot([95,105],[125,150],"r--")

plt.show() # affichage du graphique
plt.close()

```



In [15]:

```

from math import *
# from random import *
import matplotlib.pyplot as plt
from matplotlib.patches import Ellipse

fig, ax = plt.subplots()
plt.xlim(40,125)
plt.ylim(20,200)
plt.grid()
#plt.axis('equal')

points_de_vie1 = [int(p['Points de vie']) for p in
pokemons_eau] # construction du tableau des points de vie
valeur_attaque1 = [int(p['Attaque']) for p in
pokemons_eau] # construction du tableau des valeurs
d'attaque
points_de_vie2 = [int(p['Points de vie']) for p in
pokemons_psy] # construction du tableau des points de vie
valeur_attaque2 = [int(p['Attaque']) for p in

```

```

pokemons_psy] # construction du tableau des valeurs
d'attaque

plt.plot(points_de_vie1, valeur_attaque1, 'vb') #
construction du graphique
plt.plot(points_de_vie2, valeur_attaque2, 'ro') #
construction du graphique

plt.plot(95,125, 'X') # construction du graphique
plt.xlabel('Points de vie')
plt.ylabel('Attaque')

plt.annotate('Nouvel élément à classifier', xy=(95, 125),
xytext=(70, 175), fontsize=12,
          arrowprops={'facecolor':'black',
'width':1,'shrink':0.05})

# Ajout d'un cercle (ou d'une ellipse)
#ax.add_artist(plt.Circle((97, 125), 25, edgecolor='g',
facecolor='none'))
#ax.add_artist(plt.Circle((5, 12.5), 4, edgecolor='b',
facecolor='none'))

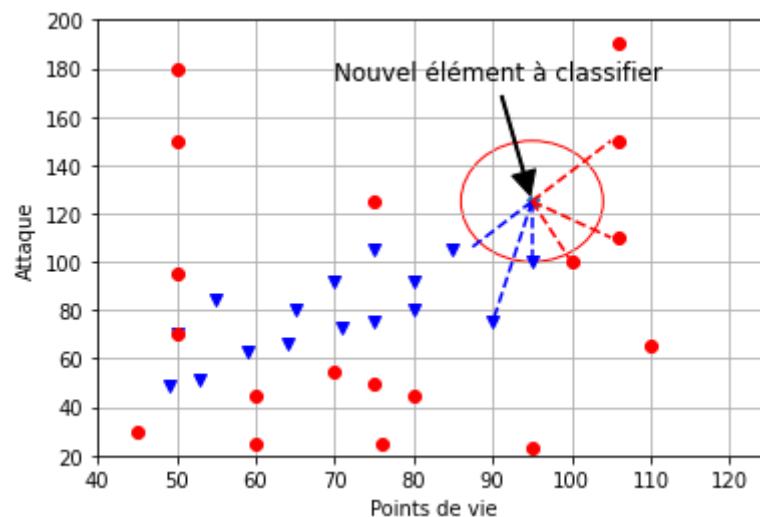
ellipse = Ellipse(xy=(95, 125), width=18, height=50,
                  edgecolor='r', fc='None', lw=1)
ax.add_patch(ellipse)

plt.plot([95,95],[125,101],"b--")
plt.plot([95,90],[125,75],"b--")
plt.plot([95,87],[125,105],"b--")

plt.plot([95,100],[125,100],"r--")
plt.plot([95,105],[125,110],"r--")
plt.plot([95,105],[125,150],"r--")

plt.show() # affichage du graphique
plt.close()

```



```
In [8]: #!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Mar 27 11:40:25 2020

@author: michael-maison
"""
from math import *
# from random import *
import matplotlib.pyplot as plt
# import numpy as np
# from scipy.stats import linregress

# Données de type 1
liste_x_1=[2,15,2,15,15,4,11,1,10,12,16,20,20,1,15,1,4,18]
liste_y_1=[12,19,5,18,7,17,9,20,10,6,5,5,1,20,7,10,12,6]

# Données de type 2
liste_x_2=[17,20,5,15,9,8,18,10,9,4,12,8,10,4,18]
liste_y_2=[20,11,9,11,12,1,18,13,5,4,16,5,12,6,5]

#liste_x_3=[12]
#liste_y_3=[12.5]

liste_x_3=[5]
liste_y_3=[12.5]

fig, ax = plt.subplots()

plt.axis([0,22, 0, 22]) # Attention [x1,x2,y1,y2]
plt.xticks( [2.5 ,5, 7.5, 10, 12.5, 15, 17.5, 20 ])#
plt.xlabel('Force')
plt.ylabel('Courage')
plt.title('Représentation des personnages')
#plt.axis('equal') # pour avoir un repère orthonormé. Faire
```

```

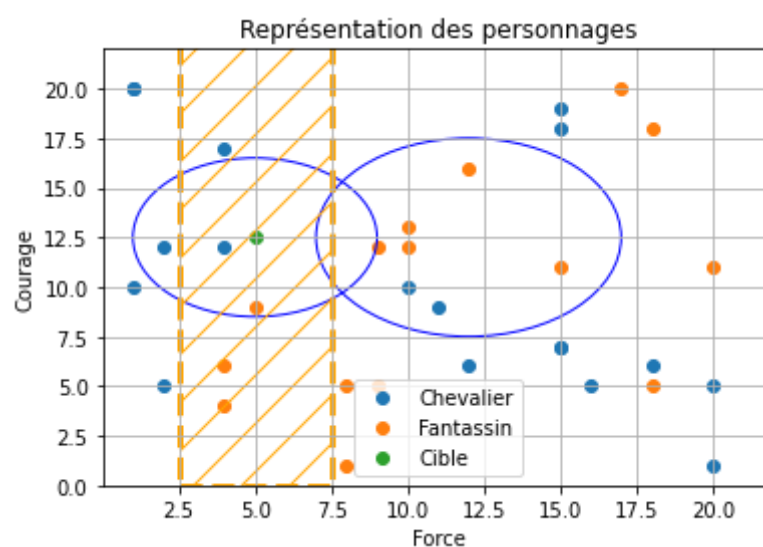
des tests.
plt.grid()
plt.scatter(liste_x_1,liste_y_1, label='Chevalier')
plt.scatter(liste_x_2,liste_y_2, label='Fantassin')
plt.scatter(liste_x_3,liste_y_3, label='Cible')
plt.legend()

# Ajout d'un cercle (ou d'une ellipse)
ax.add_artist(plt.Circle((12, 12.5), 5, edgecolor='b',
facecolor='none'))
ax.add_artist(plt.Circle((5, 12.5), 4, edgecolor='b',
facecolor='none'))

# Ajout d'un rectangle Rectangle(xy, width, height,
angle=0.0, **kwargs)[source]
# ax.add_artist(
#     plt.Rectangle((0,10), 30, 5,
# #                 edgecolor = 'orange', facecolor =
# 'none',
# #                 fill = True, hatch = '/', linestyle =
# 'dashed',
# #                 linewidth = 3, zorder = 1))

ax.add_artist(
    plt.Rectangle((2.5,0), 5, 30,
        edgecolor = 'orange', facecolor =
'none',
        fill = True, hatch = '/', linestyle =
'dashed',
        linewidth = 3, zorder = 1))
plt.show()

```



In []:



