

# Notebook\_chapitre\_6\_Visualisation\_avec\_Matplotlib

February 1, 2023

## 1 3. Créer son premier graphique simple

### 1.1 3.1. Préparer son jeu de données

```
[1]: import pandas as pd
donnees=pd.read_csv("../datasets/
↳120-years-of-olympic-history-athletes-and-results.csv", index_col=[1])
donnees.isna().sum()
```

```
[1]: ID          0
Sex           0
Age          9474
Height       60171
Weight       62875
Team          0
NOC           0
Games        0
Year          0
Season        0
City          0
Sport         0
Event         0
Medal        231333
dtype: int64
```

```
[2]: import pandas as pd
donnees=pd.read_csv("../datasets/
↳120-years-of-olympic-history-athletes-and-results.csv", index_col=[1])
donnees['Medal']=donnees['Medal'].fillna("None")
donnees_NaN_supprimes = donnees.dropna(how="any")
donnees[['Age', 'Height', 'Weight']] = donnees[['Age', 'Height', 'Weight']].
↳fillna(donnees[['Age', 'Height', 'Weight']].mean())
donnees_NaN_replaces=donnees.drop_duplicates()
donnees_NaN_supprimes=donnees_NaN_supprimes.drop_duplicates()
print("Tableau avec NaN remplacés")
print(donnees_NaN_replaces.shape)
print("Tableau avec suppression des lignes contenant des NaN")
```

```
print(donnees_NaN_supprimes.shape)
```

Tableau avec NaN remplacés

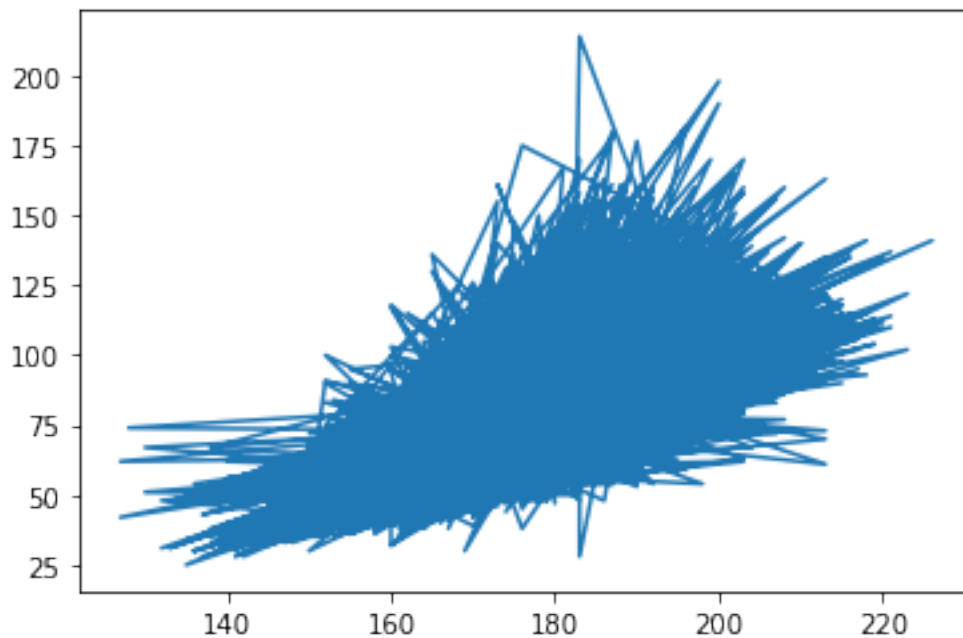
(269731, 14)

Tableau avec suppression des lignes contenant des NaN

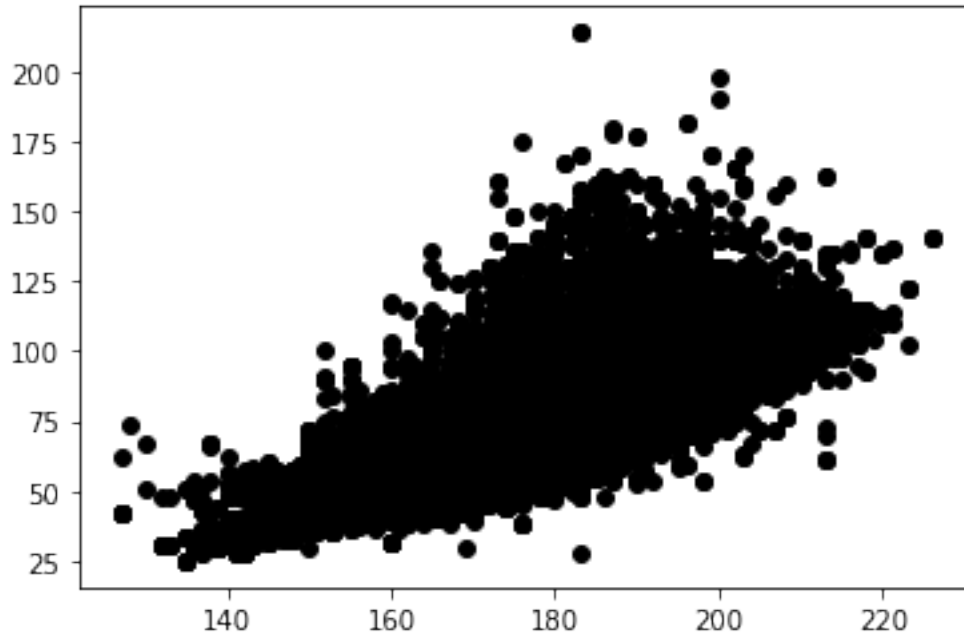
(206152, 14)

## 1.2 3.2. Créer un nuage de points

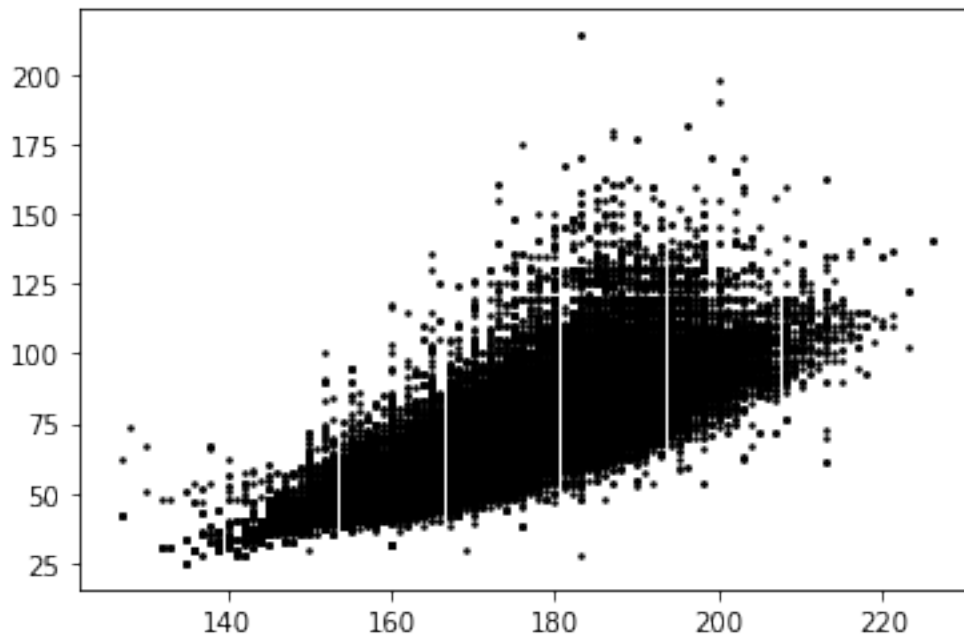
```
[3]: import matplotlib.pyplot as plt
plt.figure()
plt.subplot(111)
plt.plot(donnees_NaN_supprimes["Height"], donnees_NaN_supprimes["Weight"])
plt.show()
```



```
[4]: import matplotlib.pyplot as plt
plt.figure()
plt.subplot(111)
plt.
    ↪ plot(donnees_NaN_supprimes["Height"], donnees_NaN_supprimes["Weight"], marker='o',
    ↪ color="k", linestyle='')
plt.show()
```

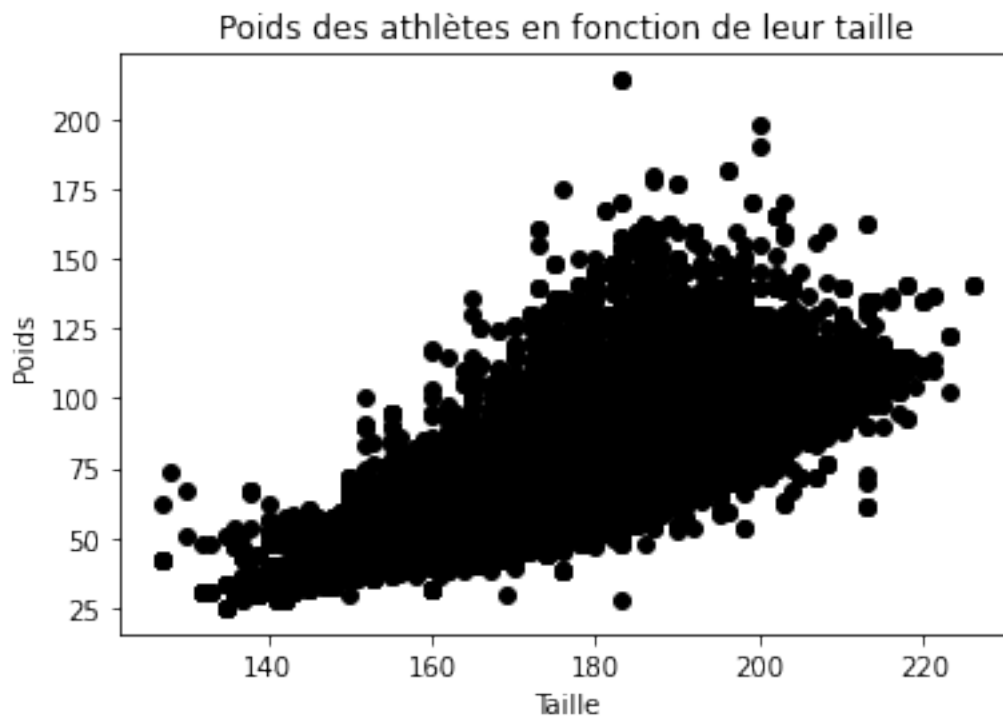


```
[5]: import matplotlib.pyplot as plt
plt.figure()
plt.subplot(111)
plt.
    ↪ plot(donnees_NaN_supprimes["Height"], donnees_NaN_supprimes["Weight"], marker='o',
    ↪ color="black", linestyle='', markersize=1.5)
plt.show()
```



### 1.3 3.3. Ajouter un titre principal et des labels aux axes du nuage de points

```
[6]: import matplotlib.pyplot as plt
plt.figure()
plt.subplot(111)
plt.
    ↪plot(donnees_NaN_supprimes["Height"], donnees_NaN_supprimes["Weight"], marker='o',
    ↪color="black", linestyle='')
plt.title("Poids des athlètes en fonction de leur taille")
plt.ylabel("Poids")
plt.xlabel("Taille")
plt.show()
```



### 1.4 3.4. Enregistrer son graphique

```
[7]: import matplotlib.pyplot as plt
plt.figure()
plt.subplot(111)
```

```

plt.
↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',↳
↳color="black", linestyle='')
plt.title("Poids en fonction de la taille des athlètes des JOs")
plt.ylabel("Poids")
plt.xlabel("Taille")
plt.savefig("athlètes.png", dpi=300, format="png")
plt.close()

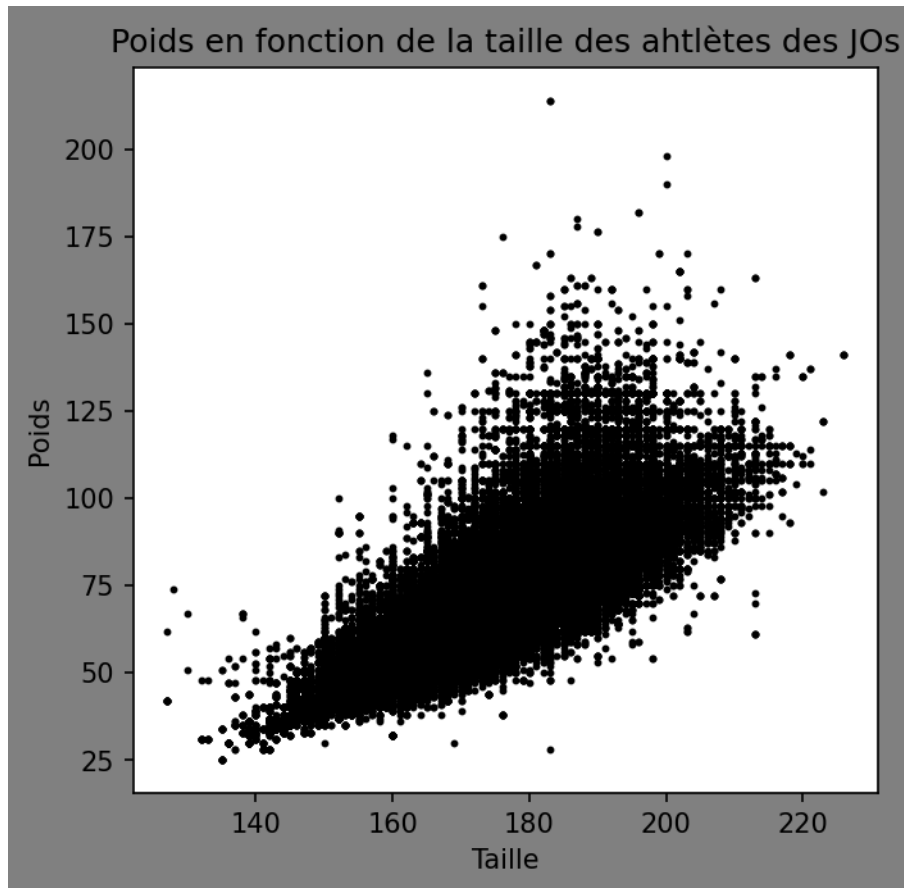
```

### 1.5 3.5. Changer la taille de la fenêtre graphique et la résolution de son graphique

```

[8]: import matplotlib.pyplot as plt
plt.figure(figsize=[5,5], dpi=150, facecolor='grey')
plt.subplot(111)
plt.
↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',↳
↳color="black", linestyle='', markersize=1.8)
plt.title("Poids en fonction de la taille des athlètes des JOs")
plt.ylabel("Poids")
plt.xlabel("Taille")
plt.show()

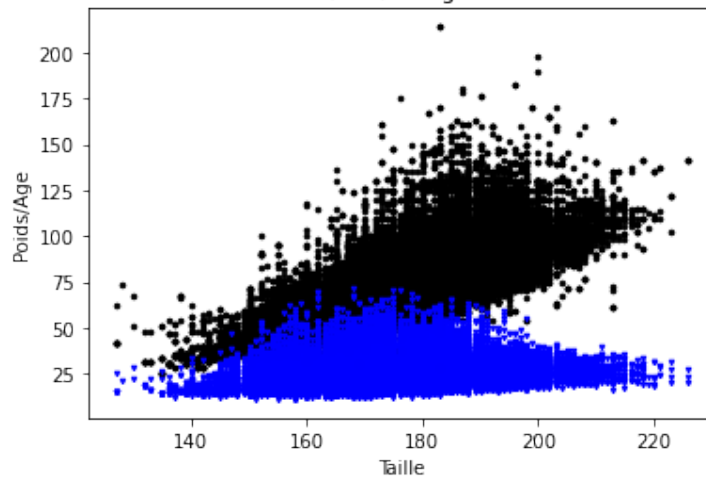
```



### 1.6 3.6. Tracer plusieurs courbes sur un même graphique (sur un même objet axes)

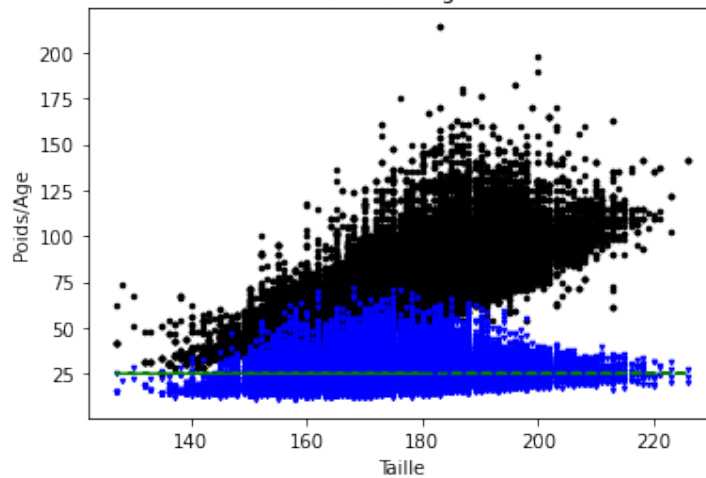
```
[9]: import matplotlib.pyplot as plt
plt.figure()
plt.subplot(111)
plt.
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',
    ↳color="black", linestyle='', markersize=2.5)
plt.
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Age"],marker='v',
    ↳color="blue", linestyle='', markersize=2.5)
plt.title("Poids en fonction de la taille des athlètes (noir) et Age en
    ↳fonction de la taille des athlètes (bleu)")
plt.ylabel("Poids/Age")
plt.xlabel("Taille")
plt.show()
```

Poids en fonction de la taille des athlètes (noir) et Age en fonction de la taille des athlètes (bleu)



```
[10]: import matplotlib.pyplot as plt
import numpy as np
plt.figure()
plt.subplot(111)
plt.
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',↳
    ↳color="black", linestyle='', markersize=2.5)
plt.
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Age"],marker='v',↳
    ↳color="blue", linestyle='', markersize=2.5)
plt.title("Poids en fonction de la taille des athlètes (noir) et Age en↳
    ↳fonction de la taille des athlètes (bleu)")
plt.plot(donnees_NaN_supprimes["Height"],[np.
    ↳mean(donnees_NaN_supprimes["Age"])]*len(donnees_NaN_supprimes["Height"]),↳
    ↳linestyle='--', color="green")
plt.ylabel("Poids/Age")
plt.xlabel("Taille")
plt.show()
```

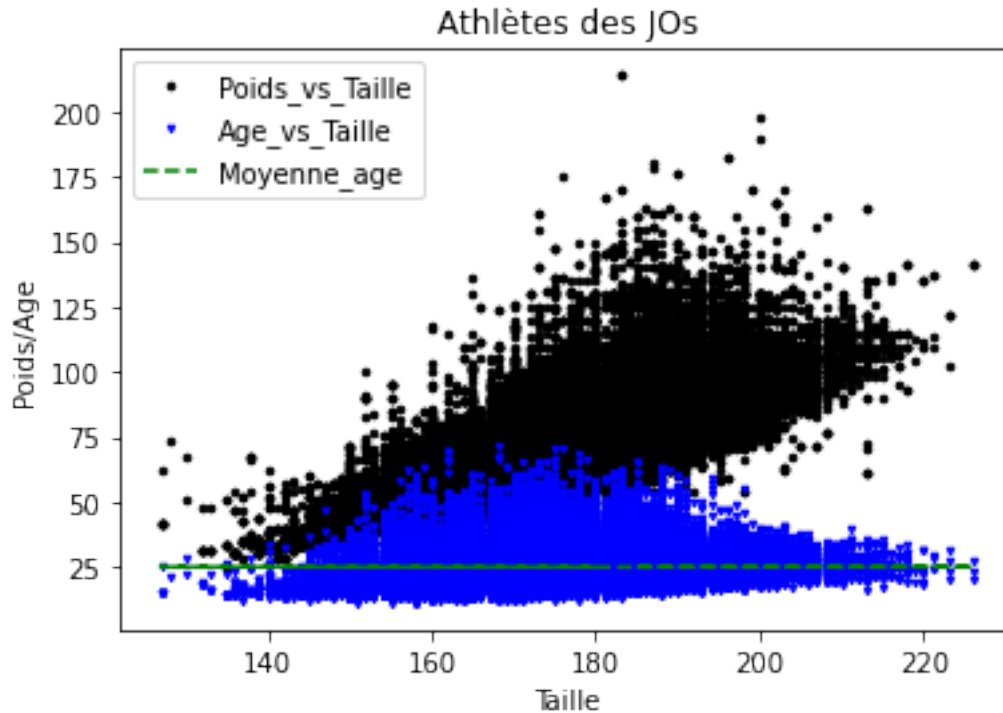
Poids en fonction de la taille des athlètes (noir) et Age en fonction de la taille des athlètes (bleu)



### 1.7 3.7. Ajouter une légende à son graphique

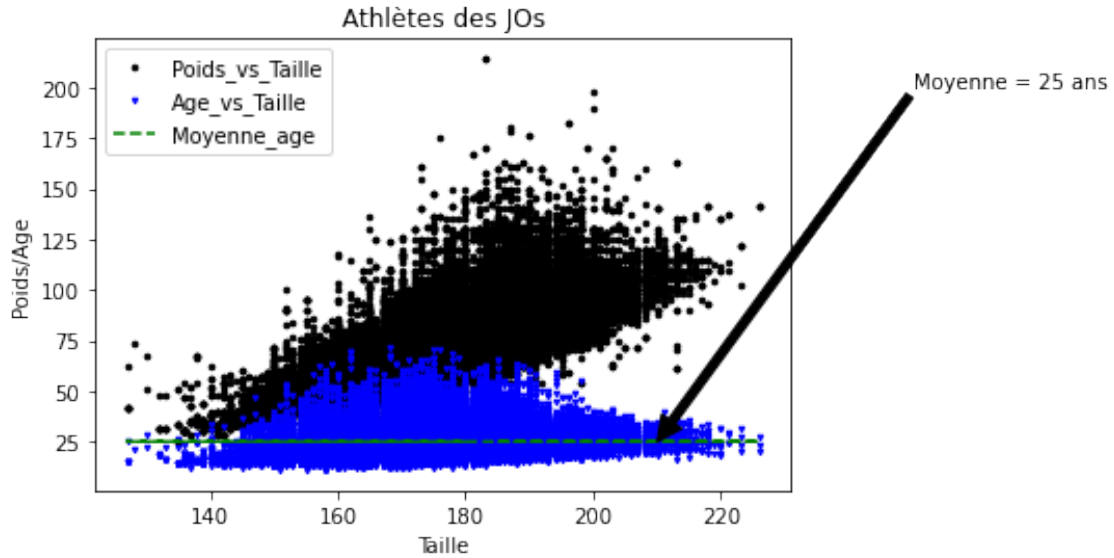
```
[11]: import matplotlib.pyplot as plt
import numpy as np
plt.figure()
plt.subplot(111)
plt.
    ↪plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',
    ↪color="black", linestyle='', markersize=2.5, label='Poids_vs_Taille')
plt.
    ↪plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Age"],marker='v',
    ↪color="blue", linestyle='', markersize=2.5, label='Age_vs_Taille')
plt.plot(donnees_NaN_supprimes["Height"],[np.
    ↪mean(donnees_NaN_supprimes["Age"])*len(donnees_NaN_supprimes["Height"]),
    ↪linestyle='--', color="green",label='Moyenne_age')
plt.title("Athlètes des JOs")
plt.ylabel("Poids/Age")
plt.xlabel("Taille")
plt.legend(loc='upper left')
plt.show()
```



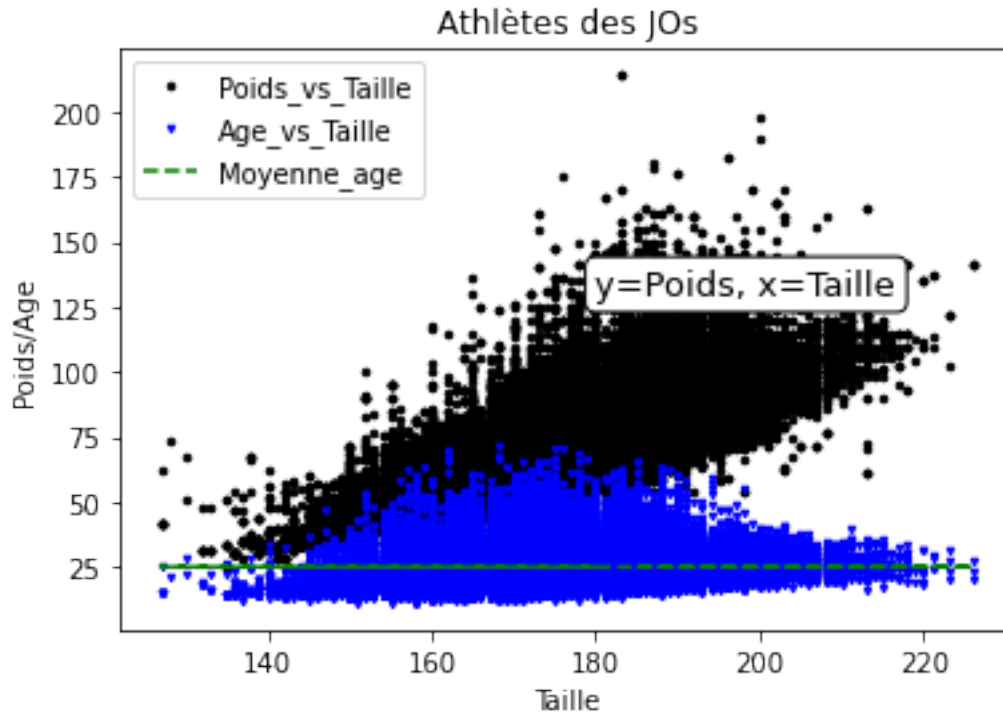


### 1.8 3.8. Annoter son graphique avec du texte

```
[12]: import matplotlib.pyplot as plt
import numpy as np
plt.figure()
plt.subplot(111)
plt.
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',
    ↳color="black", linestyle='', markersize=2.5, label='Poids_vs_Taille')
plt.
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Age"],marker='v',
    ↳color="blue", linestyle='', markersize=2.5, label='Age_vs_Taille')
plt.plot(donnees_NaN_supprimes["Height"],[np.
    ↳mean(donnees_NaN_supprimes["Age"])]*len(donnees_NaN_supprimes["Height"]),
    ↳linestyle='--', color="green",label='Moyenne_age')
plt.annotate("Moyenne = 25 ans", xy=(210,25),
    ↳xytext=(250,200),arrowprops=dict(facecolor='black'))
plt.title("Athlètes des JOs")
plt.ylabel("Poids/Age")
plt.xlabel("Taille")
plt.legend(loc='upper left')
plt.show()
```



```
[13]: import matplotlib.pyplot as plt
import numpy as np
plt.figure()
plt.subplot(111)
plt.
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',
    ↳color="black", linestyle='', markersize=2.5, label='Poids_vs_Taille')
plt.
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Age"],marker='v',
    ↳color="blue", linestyle='', markersize=2.5, label='Age_vs_Taille')
plt.plot(donnees_NaN_supprimes["Height"],[np.
    ↳mean(donnees_NaN_supprimes["Age"])*len(donnees_NaN_supprimes["Height"]),
    ↳linestyle='--', color="green",label='Moyenne_age')
plt.text(180,130,"y=Poids, x=Taille",size=13, bbox=dict(boxstyle="round",
    ↳fc="white", ec="k"))
plt.title("Athlètes des JOs")
plt.ylabel("Poids/Age")
plt.xlabel("Taille")
plt.legend(loc='upper left')
plt.show()
```

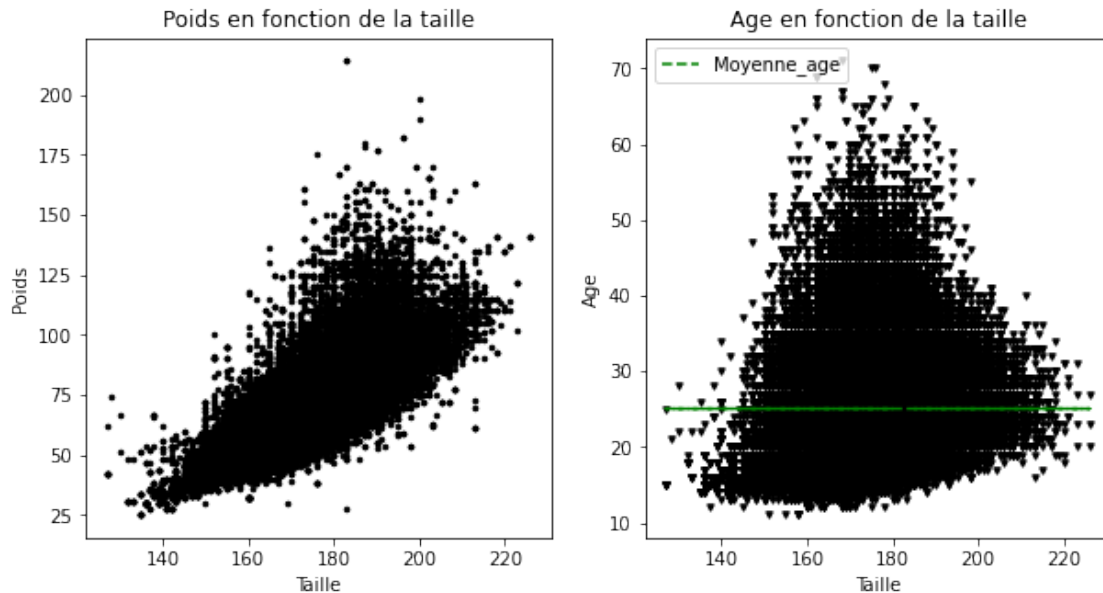


## 1.9 3.9. Combiner plusieurs graphiques grâce à subplots

### 1.9.1 3.9.1. Tracer des sous-graphiques (subplot) sur une ligne ou une colonne

```
[14]: import matplotlib.pyplot as plt
import numpy as np
plt.figure(figsize=[10,5])
plt.subplot(121)
plt.
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',
    ↳color="black", linestyle='', markersize=2.5)
plt.title("Poids en fonction de la taille")
plt.ylabel("Poids")
plt.xlabel("Taille")
plt.subplot(122)
plt.
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Age"],marker='v',
    ↳color="black", linestyle='', markersize=3.5)
plt.plot(donnees_NaN_supprimes["Height"],[np.
    ↳mean(donnees_NaN_supprimes["Age"])]*len(donnees_NaN_supprimes["Height"]),
    ↳linestyle='--', color="green",label='Moyenne_age')
plt.title("Age en fonction de la taille")
plt.ylabel("Age")
```

```
plt.xlabel("Taille")
plt.legend(loc='upper left')
plt.show()
```



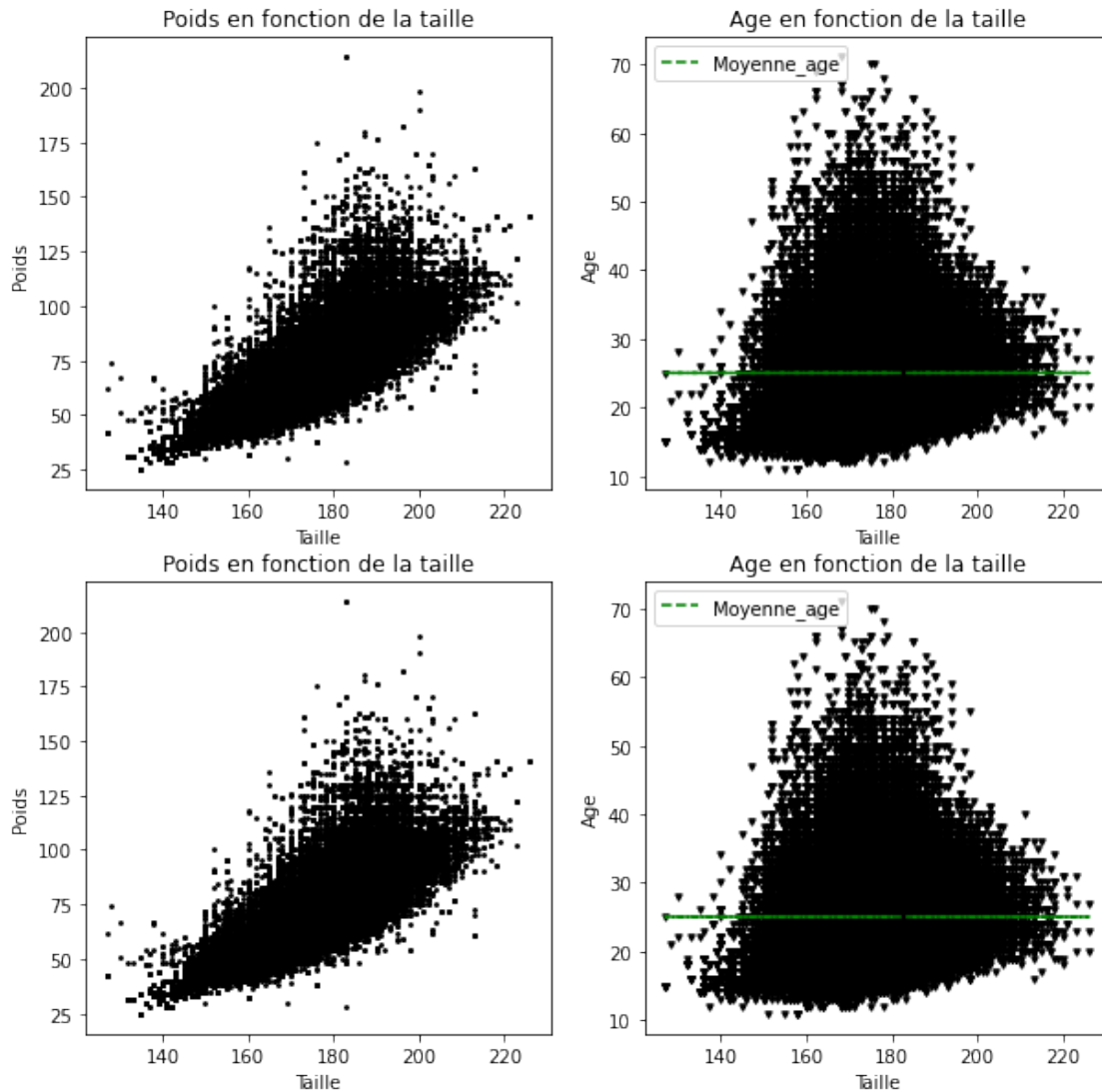
### 1.9.2 3.9.2. Tracer des sous-graphiques (subplots) sur plusieurs lignes et plusieurs colonnes

```
[15]: import matplotlib.pyplot as plt
import numpy as np
fig, axs = plt.subplots(2, 2, figsize=(10, 10))
axs[0,0].
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',
    ↳color="black", linestyle='', markersize=1.8)
axs[0,0].set_title("Poids en fonction de la taille")
axs[0,0].set_ylabel("Poids")
axs[0,0].set_xlabel("Taille")
axs[0,1].
    ↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Age"],marker='v',
    ↳color="black", linestyle='', markersize=3.5)
axs[0,1].plot(donnees_NaN_supprimes["Height"],[np.
    ↳mean(donnees_NaN_supprimes["Age"])]*len(donnees_NaN_supprimes["Height"]),
    ↳linestyle='--', color="green",label='Moyenne_age')
axs[0,1].set_title("Age en fonction de la taille")
axs[0,1].set_ylabel("Age")
axs[0,1].set_xlabel("Taille")
axs[0,1].legend(loc='upper left')
```

```

axs[1,0].
↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',
↳color="black", linestyle='', markersize=1.8)
axs[1,0].set_title("Poids en fonction de la taille")
axs[1,0].set_ylabel("Poids")
axs[1,0].set_xlabel("Taille")
axs[1,1].
↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Age"],marker='v',
↳color="black", linestyle='', markersize=3.5)
axs[1,1].plot(donnees_NaN_supprimes["Height"],[np.
↳mean(donnees_NaN_supprimes["Age"])]*len(donnees_NaN_supprimes["Height"]),
↳linestyle='--', color="green",label='Moyenne_age')
axs[1,1].set_title("Age en fonction de la taille")
axs[1,1].set_ylabel("Age")
axs[1,1].set_xlabel("Taille")
axs[1,1].legend(loc='upper left')
plt.show()

```



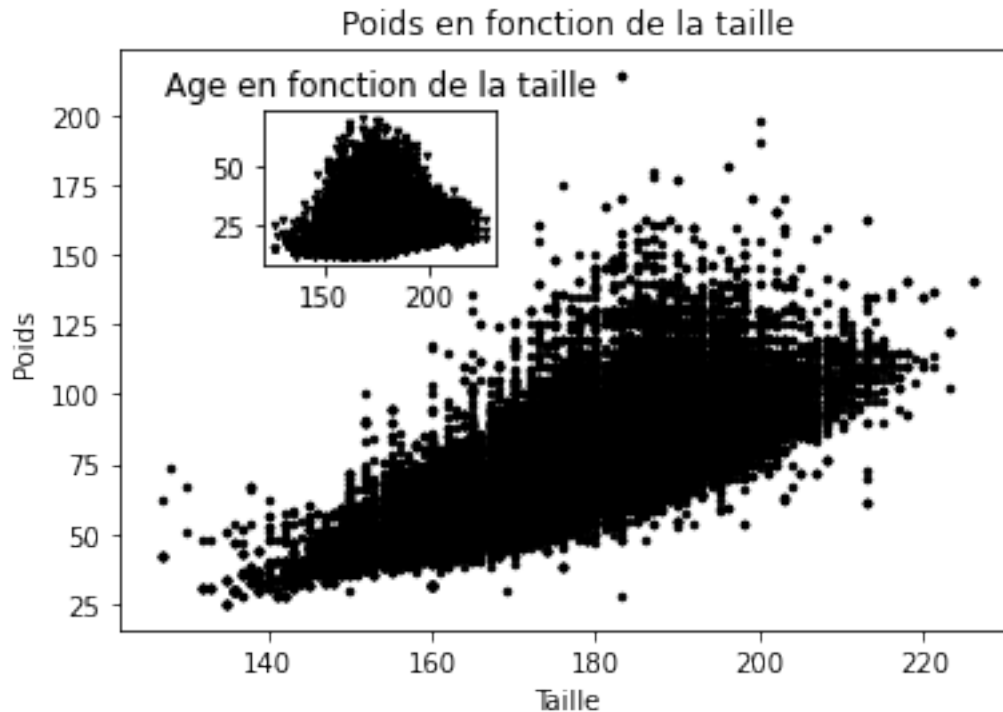
### 1.9.3 3.9.3. Incruster un objet axes dans un autre

```
[16]: import matplotlib.pyplot as plt
fig, axe_principal = plt.subplots()
axe_principal.
    ↪ plot(donnees_NaN_supprimes["Height"], donnees_NaN_supprimes["Weight"], marker='o',
    ↪ color="black", linestyle='', markersize=2.5)
axe_principal.set_title("Poids en fonction de la taille")
axe_principal.set_ylabel("Poids")
axe_principal.set_xlabel("Taille")
axe_incruste=fig.add_axes([0.25,0.6,.2,.2])
```

```

axe_incruste.
↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Age"],marker='v',
↳color="black", linestyle='', markersize=2.5)
axe_incruste.set_title("Age en fonction de la taille")
plt.show()

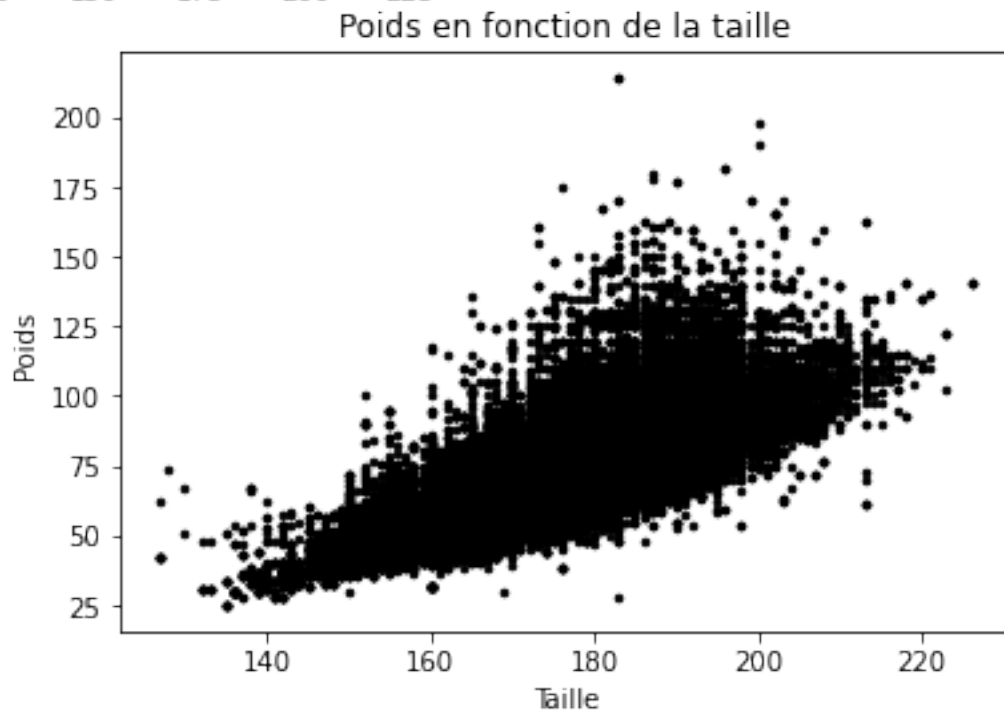
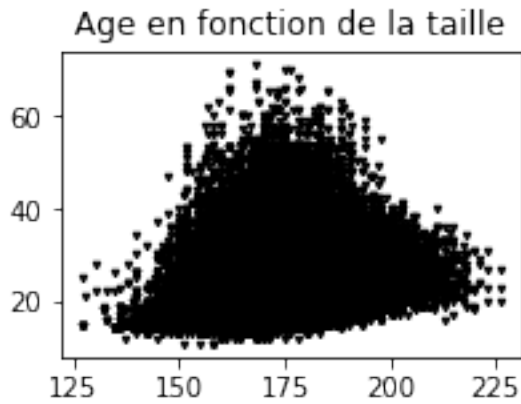
```



```

[17]: import matplotlib.pyplot as plt
fig, axe_principal = plt.subplots()
axe_principal.
↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Weight"],marker='o',
↳color="black", linestyle='', markersize=2.5)
axe_principal.set_title("Poids en fonction de la taille")
axe_principal.set_ylabel("Poids")
axe_principal.set_xlabel("Taille")
axe_incruste=fig.add_axes([0,1,.4,.4])
axe_incruste.
↳plot(donnees_NaN_supprimes["Height"],donnees_NaN_supprimes["Age"],marker='v',
↳color="black", linestyle='', markersize=2.5)
axe_incruste.set_title("Age en fonction de la taille")
plt.show()

```

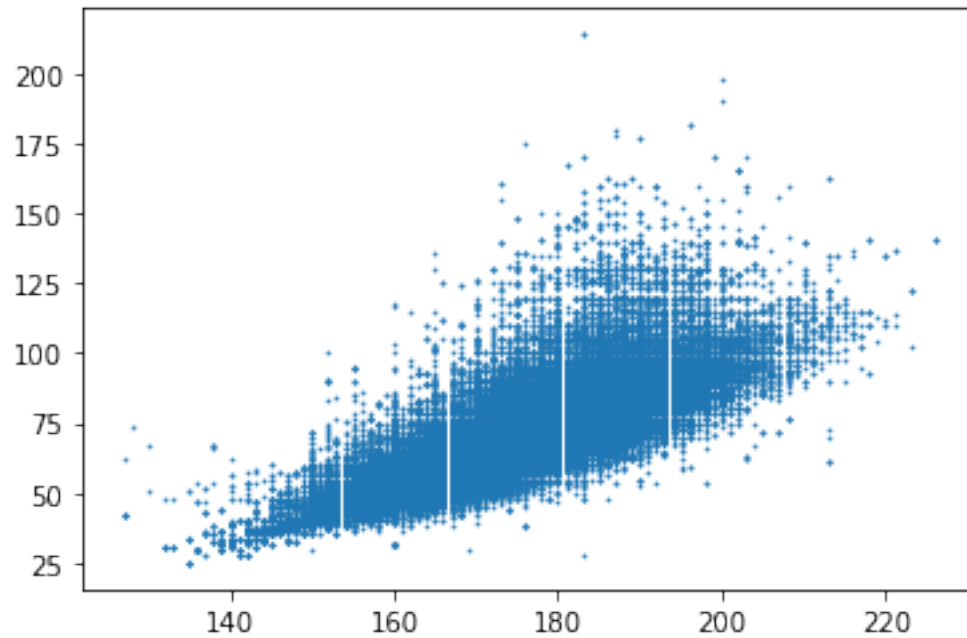


## 2 4. Les différents types de graphes

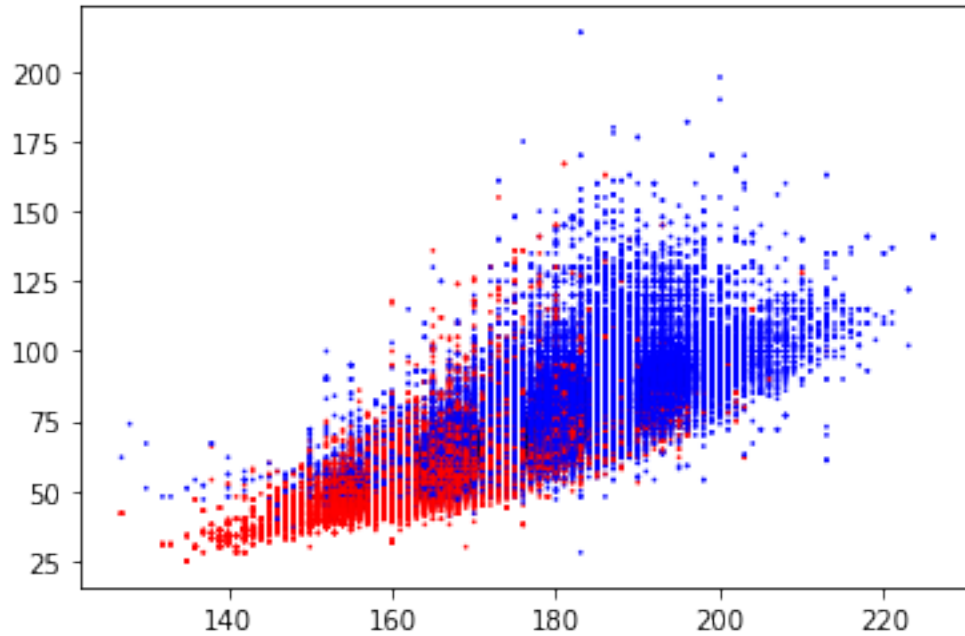
### 2.1 4.2. Scatterplot

```
[18]: import matplotlib.pyplot as plt
plt.figure()
plt.subplot(111)
plt.scatter(donnees_NaN_supprimes["Height"],
↳ donnees_NaN_supprimes["Weight"],s=0.5)
plt.show()
```





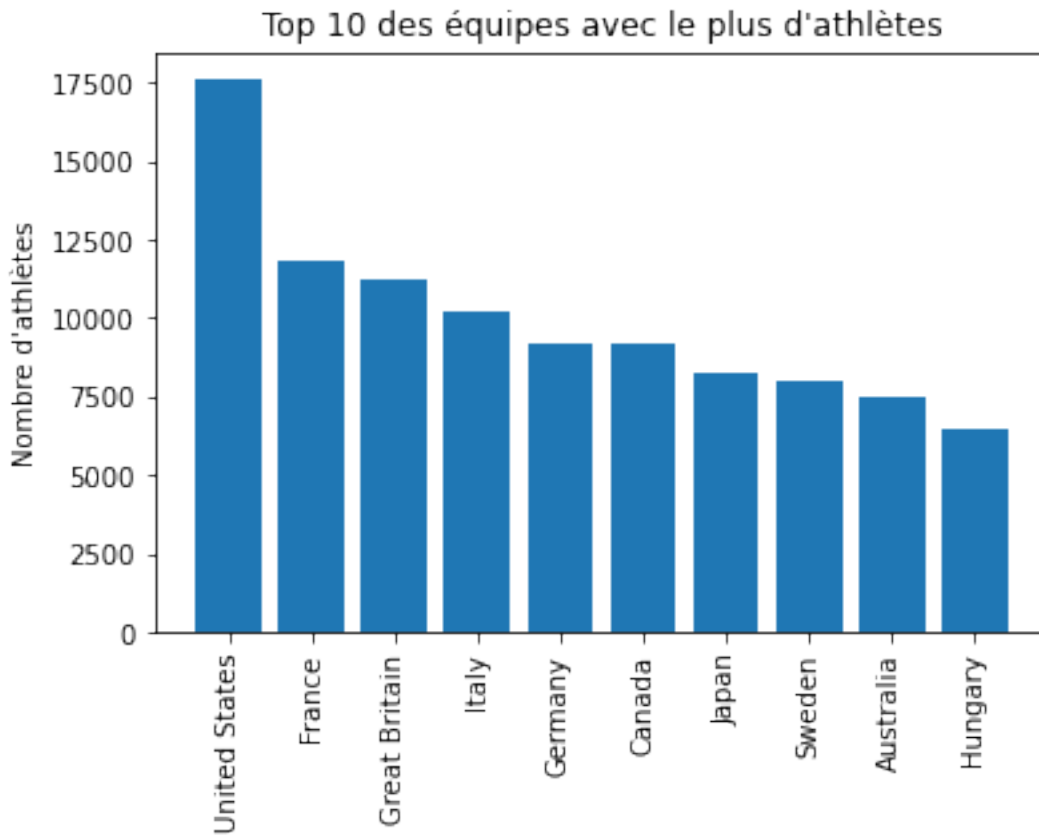
```
[19]: import matplotlib.pyplot as plt
plt.figure()
plt.subplot(111)
colors = {'F': 'red', 'M': 'blue'}
plt.scatter(donnees_NaN_supprimes["Height"],
            ↪ donnees_NaN_supprimes["Weight"], c=donnees_NaN_supprimes['Sex'].apply(lambda
            ↪ x: colors[x]), s=0.5)
plt.show()
```



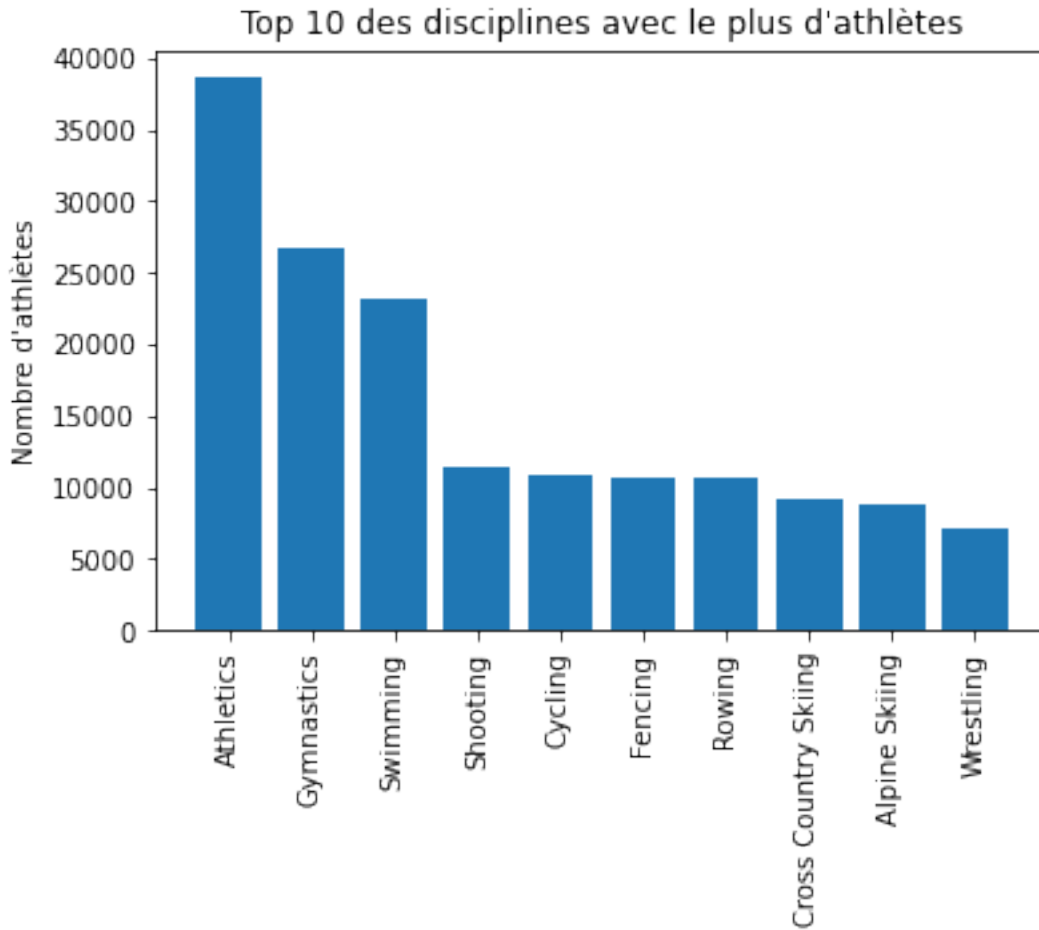
## 2.2 4.3. Graphique à barres (bargraph)

### 2.2.1 4.3.1. Graphique à barres simple

```
[20]: import matplotlib.pyplot as plt
import pandas as pd
plt.figure()
plt.subplot(111)
labels_X=donnees_NaN_replaces['Team'].value_counts()[:10].keys()
valeurs_Y=donnees_NaN_replaces['Team'].value_counts()[:10].values
plt.bar(labels_X, valeurs_Y)
plt.xticks(labels_X,rotation=90)
plt.ylabel("Nombre d'athlètes")
plt.title("Top 10 des équipes avec le plus d'athlètes")
plt.show()
```



```
[21]: import matplotlib.pyplot as plt
import pandas as pd
plt.figure()
plt.subplot(111)
labels_X=donnees_NaN_replaces['Sport'].value_counts()[:10].keys()
valeurs_Y=donnees_NaN_replaces['Sport'].value_counts()[:10].values
plt.bar(labels_X, valeurs_Y)
plt.xticks(labels_X,rotation=90)
plt.ylabel("Nombre d'athlètes")
plt.title("Top 10 des disciplines avec le plus d'athlètes")
plt.show()
```



#### 2.2.2 4.3.2. Graphique à barres groupées

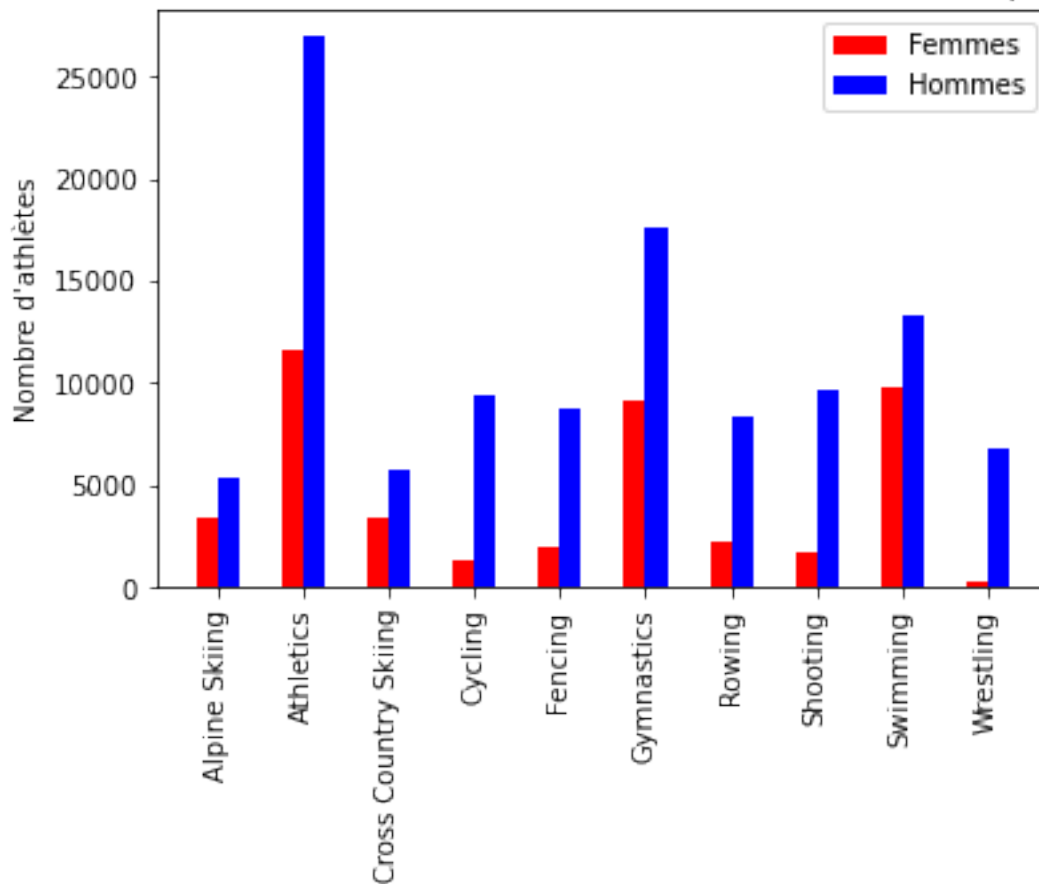
```
[22]: sports_10=donnees_NaN_replaces['Sport'].value_counts()[:10].keys()
donnees_bar=donnees_NaN_replaces.loc[donnees_NaN_replaces['Sport'].
↳ isin(sports_10)].groupby(["Sport", "Sex"]).size().unstack(1)
donnees_bar
```

```
[22]: Sex          F          M
Sport
Alpine Skiing    3398    5431
Athletics        11666   26958
Cross Country Skiing 3385    5748
Cycling          1394    9433
Fencing          2000    8735
Gymnastics       9129   17578
Rowing           2193    8402
Shooting         1724    9724
```

Swimming	9850	13345
Wrestling	304	6850

```
[23]: import matplotlib.pyplot as plt
pos = list(range(len(donnees_bar['F'])))
width = 0.25
plt.figure()
plt.subplot(111)
# Créer une barre avec les données F,
# à la position pos
plt.bar(pos, donnees_bar['F'], width, color='red',label="Femmes")
# Créer une barre avec les données F,
# à la position pos + 0.25 afin que la barre M soit décalée à droite de la
↳ première barre F
plt.bar([p + width for p in pos], donnees_bar['M'],width, color='blue',↳
↳ label="Hommes")
plt.xticks([p + 0.125 for p in pos],donnees_bar.index,rotation=90)
plt.ylabel("Nombre d'athlètes")
plt.legend()
plt.title("Nombre d'athlètes hommes et femmes dans les 10 disciplines")
plt.show()
```

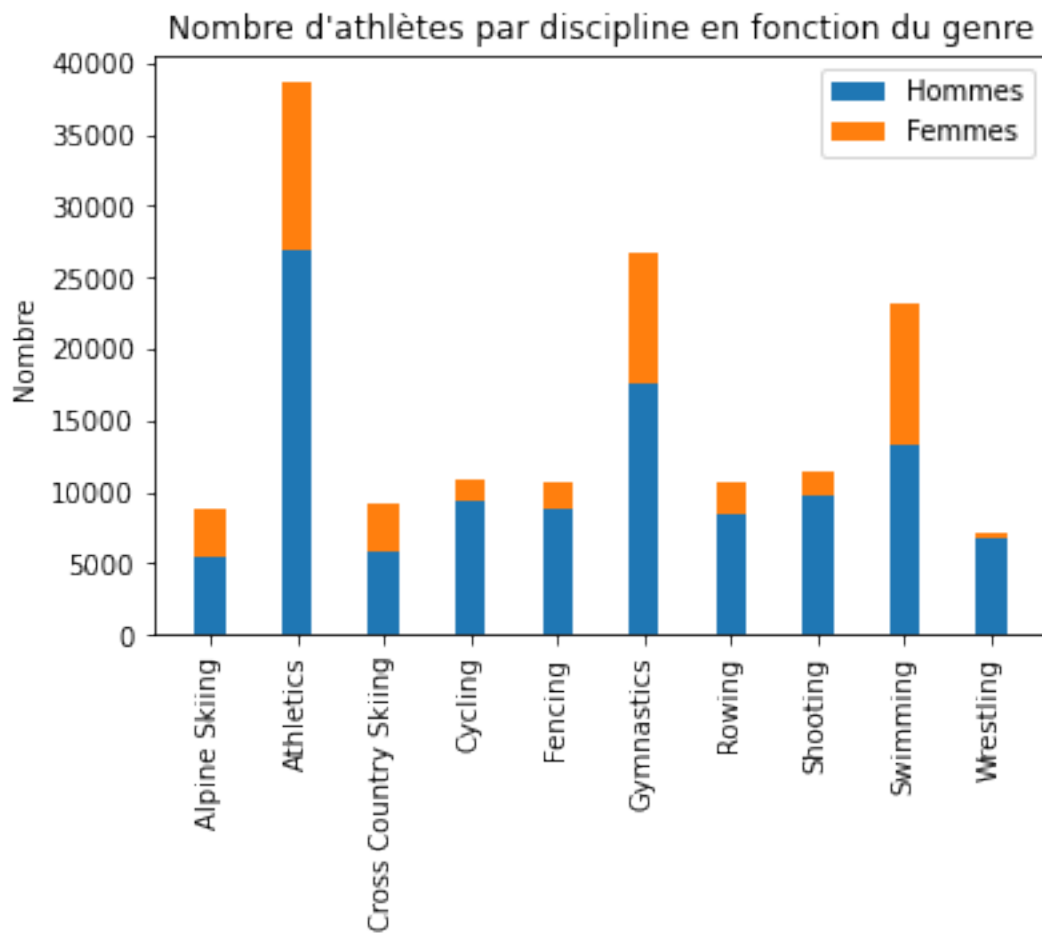
Nombre d'athlètes hommes et femmes dans les 10 disciplines



### 2.2.3 4.3.3. Graphique à barres empilées

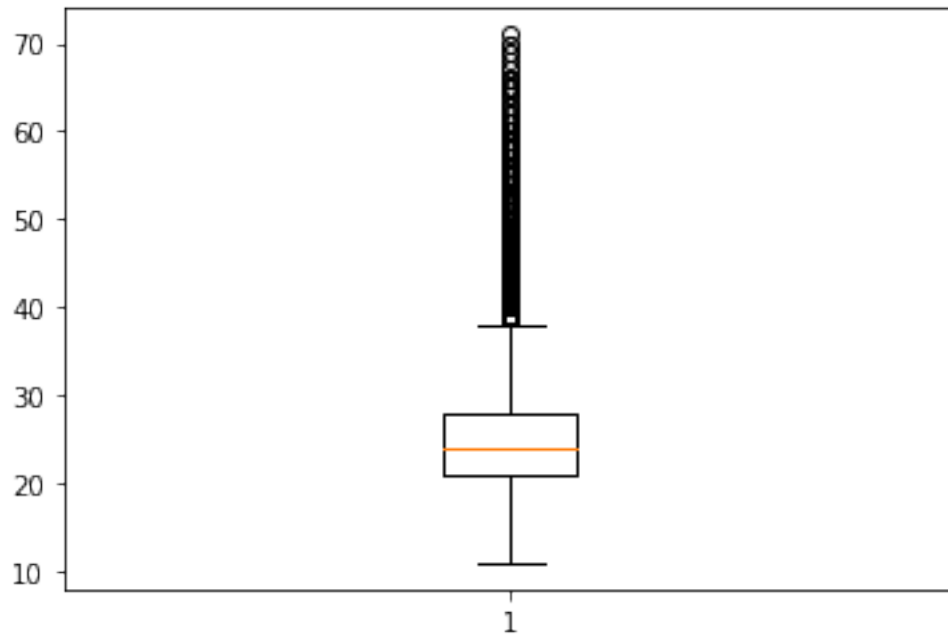
```
[24]: import matplotlib.pyplot as plt
sports_10= donnees_NaN_replaces['Sport'].value_counts()[:10].keys()
donnees_bar=donnees_NaN_replaces.loc[donnees_NaN_replaces['Sport'].
↳isin(sports_10)].groupby(["Sport", "Sex"]).size().unstack(1)
valeurs_hommes = donnees_bar["M"]
valeurs_femmes = donnees_bar["F"]
width = 0.35
plt.figure()
plt.subplot(111)
plt.bar(donnees_bar.index, valeurs_hommes, width, label='Hommes')
plt.bar(donnees_bar.index, valeurs_femmes, width,
↳bottom=valeurs_hommes, label='Femmes')
plt.ylabel('Nombre')
plt.title("Nombre d'athlètes par discipline en fonction du genre")
plt.xticks(donnees_bar.index,rotation=90)
```

```
plt.legend()  
plt.show()
```



## 2.3 4.4. Box plots

```
[25]: import matplotlib.pyplot as plt  
plt.figure()  
plt.subplot(111)  
plt.boxplot(donnees_NaN_supprimes["Age"])  
plt.show()
```



```
[26]: import matplotlib.pyplot as plt
plt.figure()
plt.subplot(111)
plt.boxplot(donnees_NaN_supprimes["Age"],showfliers=False)
plt.title("Boite à moustache sur la variable Age")
plt.xticks([1], ['Age'])
plt.show()
```



Boite à moustache sur la variable Age

