

Code de partage avec Capytale : 1680-1272508

[Corrigé](#)

## Préparation

Dans un premier temps, nous allons à nouveau importer le jeu de données. Sur Capytale ou dans le fichier Python, on écrira et on exécutera :

```
import pandas
films=pandas.read_csv('films.csv', delimiter=';', encoding='utf8')
```

## Représenter des données avec des diagrammes en bâtons

### 1 Préambule : vers la représentation graphique

Pour représenter des données sous la forme de diagramme en bâtons, on utilisera la commande `plt.bar` de la bibliothèque `matplotlib`

Comme pour la représentation de fonctions, une particularité de cette commande est qu'elle nécessite deux listes : une pour les abscisses et une pour les ordonnées. On pourra tester avec le programme ci-dessous.

```
import matplotlib.pyplot as plt

valeurs = [7, 58, 49, 32]
abscisses = [1, 2, 3, 4]
plt.bar(abscisses, valeurs)
plt.show()
```

On peut aussi définir des abscisses qui ne sont pas des nombres :

```
valeurs = [7, 58, 49, 32]
abscisses = [A, B, C, D]
plt.bar(abscisses, valeurs)
plt.show()
```

### 2 Sélectionner les données

Avec le même jeu de données (films) :

1. compter le nombre de films pour une année donnée (par exemple 2004) ;

Comme nous l'avons vu lors du T.P. précédent, on peut filtrer sur l'année 2004 pour ne garder que l'extrait du tableau qui nous intéresse. Cela se fait avec la commande :

```
films[films['release_year']==2004]
```

On peut alors compter le nombre de lignes avec `.count()` mais la réponse sera donnée sous forme de tableau comportant le nombre de lignes de chaque colonne.

En sélectionnant une seule colonne, on obtient une réponse qui est sous la forme d'un nombre :

```
films[films['release_year']==2004]['release_year'].count()
```

2. calculer le revenu moyen des films sur une année (par exemple 2004).

La commande est analogue, on effectue juste la moyenne sur la colonne des données extraites :

```
films[films['release_year']==2004]['revenue'].mean()
```

*Nota bene* : on pourra utiliser les commandes `.count()` et `.mean()`

### 3 Représenter les données choisies

Représenter à l'aide de diagrammes en bâtons :

1. l'évolution du nombre de films par année;

On répète le comptage par année effectué plus haut mais en faisant varier l'année pour chaque valeur entre 1916 et 2017 (les valeurs extrêmes présentes dans le jeu de données).

Afin de représenter les données, on crée une liste qui contient les valeurs (nombre de films) pour chaque année. Puis on crée la liste des abscisses (les années) et on représente.

```
# Le nombre de films par an
nombre=[]
for i in range(1916,2017):
    nombre.append(films[films['release_year']==i]['release_year'].count())

x=[i for i in range(1916,2017)]
import matplotlib.pyplot as plt
plt.bar(x,nombre)
plt.show()
```

2. l'évolution du revenu moyen des films par année.

*Pour améliorer un peu, on pourra enlever les films dont le revenu est 0 (l'information n'est sans doute pas renseignée dans ce cas).*

Le programme est très proche, il faut juste adapter à l'information recherchée ici (le revenu moyen).

```
revenue=[]
for i in range(1916,2017):
    revenue.append(films[films['release_year']==i]['revenue'].mean())

x=[i for i in range(1916,2017)]
import matplotlib.pyplot as plt
plt.bar(x,revenue)
plt.show()
```

Pour aller plus loin, on pourra tester les paramètres suivants :

```
plt.xticks(x) # à adapter
plt.xlabel("Années")
plt.ylabel("Revenu moyen")
plt.title("Revenu moyen des films par année de 1917 à 2017")
```