T. P. n° 1
Analyse exploratoire

Résumé

Ce TP n° 1 propose un exemple d'analyse exploratoire de jeu de données.

1 Jeux vidéo : Chargement des données

Vous êtes un jeune entrepreneur désireux de faire fortune dans les jeux vidéo. Comme vous avez beaucoup d'imagination mais peu d'argent, vous souhaitez investir là où vous êtes sûr de faire du profit. Vous allez observer des données issues de ventes de jeux vidéo afin de faire votre propre étude de marché.
Cet ensemble de données contient une liste de jeux vidéo jusqu'à 2020 avec des ventes supérieures à 100 000 exemplaires. Il a été généré par le site de `vgchartz.com`.
Les champs incluent

- *Rank* : Classement des ventes globales
- *Name* : Nom du jeu
- *Platform* : Plate-forme de la version des jeux (c'est-à-dire PC, PS4, etc.)
- *Year* : Année de sortie du jeu
- *Genre* : Genre du jeu
- *Publisher* : Éditeur du jeu
- *NA_Sales* : Ventes en Amérique du Nord (en millions)
- *EU_Sales* : Ventes en Europe (en millions)
- *JP_Sales* : Ventes au Japon (en millions)
- *Other_Sales* : Ventes dans le reste du monde (en millions)
- *Global_Sales* : Total des ventes mondiales.

a) Télécharger le fichier *vgsales.csv* depuis : `https://www.kaggle.com/gregorut/videogamesales`.

b) La fonction **read_csv()** de la librairie `Pandas` lit des données de type *csv* pour les stocker dans un *dataframe*.

```
> import pandas as pd
> import numpy as np
> import matplotlib.pyplot as plt
> import seaborn as sns
> from scipy import stats
> mydata = pd.read_csv('vgsales.csv')
```

c) Il est imporatnt de pas toucher aux données originales mais de travailler sur une copie des données dans un deuxième *dataframe*. Copier le dataframe dans la variable `df` pour dataframe.

d) Observer les premières colonnes avec la fonction `df.head()`

Il est temps de distinguer une tendance sur les données ! Pour cela, la fonction **describe()** vous donne des informations sur la répartition sur toute vos variables numériques.

Travail à réaliser !

a) Cherchez et citez à quoi correspond un objet de type "*dataframe*" sous Python.

b) Afficher les types des différentes variables.

c) Appliquez la fonction `describe()` sur votre *dataframe*.

d) Vérifiez la moyenne de la colonne *JP_Sales* avec la fonction **mean()**. Comparez avec celle donnée par la fonction `describe()`.

e) A priori, quel genre a été le plus vendu ?

f) A priori, quel pays a eu les meilleures ventes ?

g) Observez la variable explicative *Year*. Pourquoi cette variable peut contredire les deux hypothèses précédentes ?

1.1 La notion de densité

Vous allez observer la densité de probabilité des données. En théorie des probabilités ou en statistique, une densité de probabilité est une fonction qui permet de représenter une loi de probabilité sous forme d'intégrales. Dans un histogramme, la densité en un point x est estimée par la proportion d'observations $x_1, x_2, ..., x_N$ qui se trouvent à proximité de x. Pour cela, nous traçons une boîte en x et dont la largeur est définie par un paramètre de lissage h (soit la largeur de la boîte) ; nous comptons ensuite le nombre d'observations qui appartiennent à cette boîte. [1]

Problème avec les histogrammes :
- nous devons définir le paramètre h
- les histogrammes produisent une estimation de la fréquence non continue.

La fonction **plot(kind = 'density')** fournit une estimation par noyau (ou encore méthode de Parzen-Rosenblatt, 1962). C'est une méthode non paramétrique d'estimation de la densité de probabilité d'une variable aléatoire. Elle se base sur un échantillon d'une population et permet d'estimer la densité de probabilité en tout point du support (intervalle min et max des valeurs observées). Cette méthode du noyau consiste à retrouver la continuité : pour cela, nous remplaçons la boîte centrée en x et de largeur h par une loi gaussienne (définie par la suite) centrée en x. Plus une observation est proche du point de support x plus la courbe en cloche lui donnera une valeur numérique importante. À l'inverse, les observations trop éloignées de x

1. Ces définitions sont inspirées de celles que vous trouvez sur Wiképédia, vous pourrez ainsi aisément retrouver la définition et la démonstration mathématique.

se voient affecter d'une valeur numérique négligeable. Notez également que plus il y a d'observations dans le voisinage d'un point, plus sa densité est élevée. La méthode du noyau est plus précise qu'un simple histogramme (fonction **hist()**).

```
> df["Global_Sales"].hist()
> plt.show()

> df["Global_Sales"].plot(kind = 'density')
> plt.show()
```

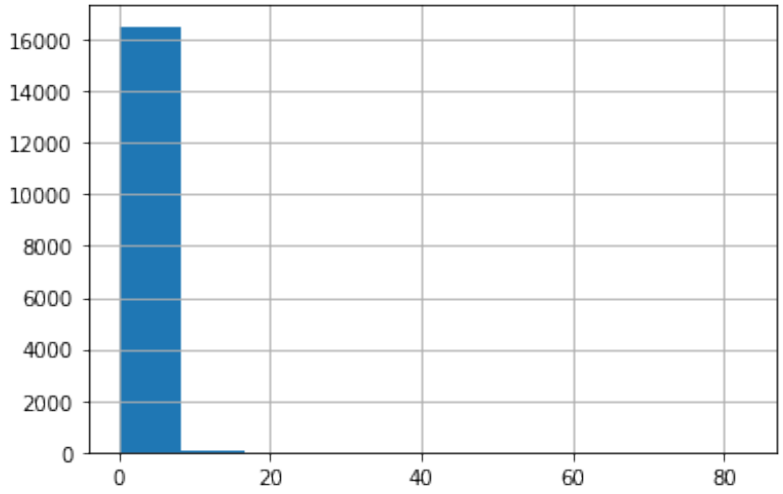

FIGURE 1 – Histogramme des valeurs de *Global_Sales*

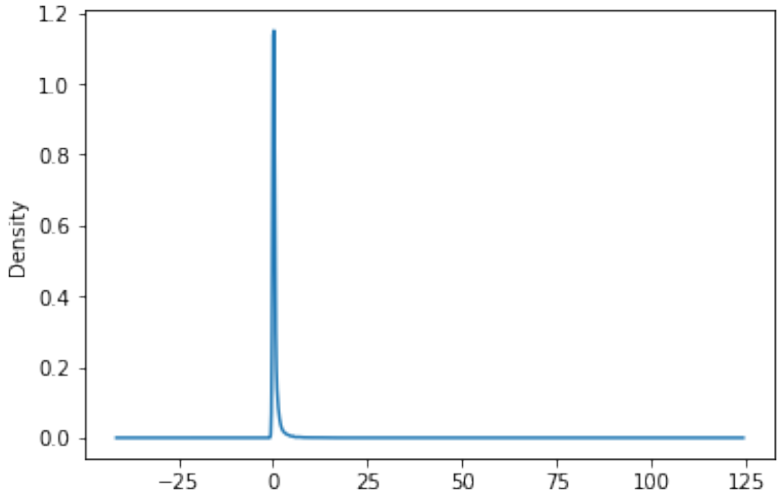

FIGURE 2 – Densité de probabilité des valeurs de *Global_Sales*

La Figure 2 n'est pas très explicite, essayons de savoir pourquoi. Le code ci-dessous permet d'afficher les *Global_Sales* pour chaque année à l'aide de la fonction **plot()**.

```
> import matplotlib.pyplot as plt
> df_temp = df.set_index('Year', inplace=False)
> df_temp['Global_Sales'].plot(legend=True, marker='.', linestyle='none')
```

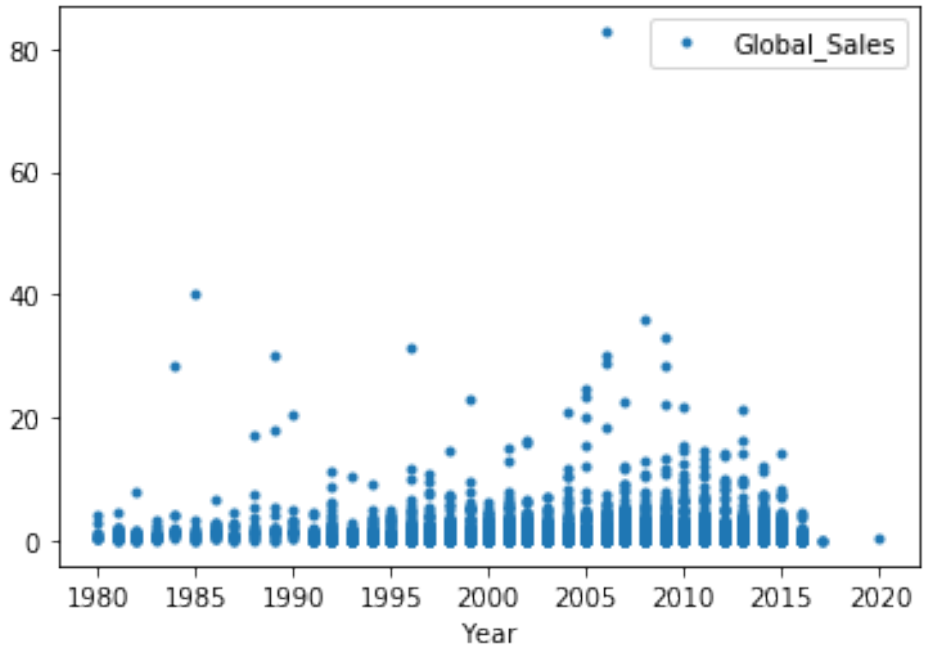

FIGURE 3 – Ventes globales par année

Travail à réaliser

a) Y-a-il une année qui se démarque des autres ? Laquelle ?

b) Pourquoi certaines valeurs extrêmes peuvent-elles fausser l'interprétation d'une variable ?

1.2 La variance

Utilisez l'option `kind='box'` de la fonction `plot` sur votre dataframe *df*.

```
> df["Global_Sales"].plot(kind='box' )
> plt.show()
```

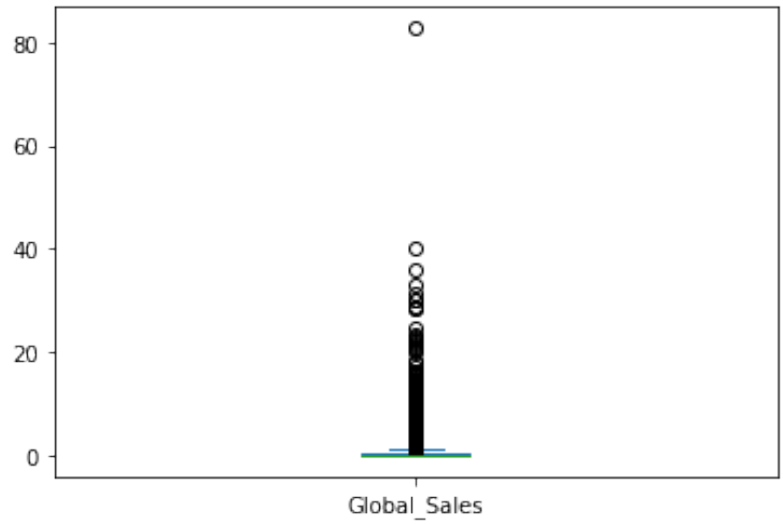

FIGURE 4 – Boxplot des *Global_Sales*

Travail à réaliser

a) Expliquez le principe de l'option **boxplot()**.

b) Expliquez en quoi une valeur perturbe potentiellement notre interprétation du résultat de **boxplot()**.

Nous allons observer les ventes européennes selon les différents genres

```
> import seaborn as sns
> sns.set(rc={'figure.figsize':(11.7,8.27)})
#create boxplot by group
> sns.boxplot(x='Genre', y='EU_Sales', data=df)
```

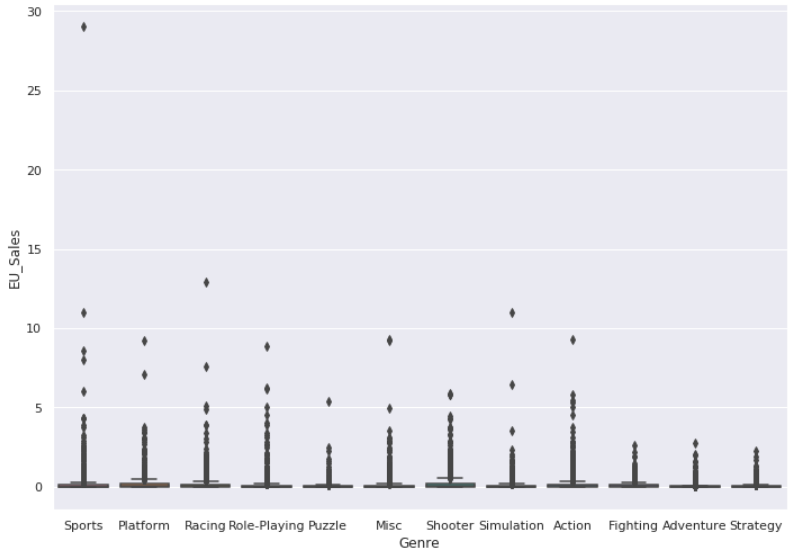

FIGURE 5 – Boxplot de *EU_Sales* par genre

Travail à réaliser

a) À quel mathématicien devons-nous la découverte de la variance ?

b) La variance permet d'obtenir l'écart type, qui est la racine carrée de la variance. Pourquoi l'écart-type est souvent plus parlant que la variance pour appréhender la dispersion ?

c) La variance est un des éléments permettant de caractériser une loi de probabilité. Pourquoi ?

Vous décidez de comparer deux genres différents : les jeux d'action et les jeux de sports.

```
> df_gs_action = df["Global_Sales"][df["Genre"]== "Action"]
> var = np.var(df_gs_action)
> print(var)
1.336920636538017
```

Travail à réaliser

a) Calculer la variance des jeux de type sport. Quelle est cette valeur ?

b) Commentez la variance du jeu *Action* et du jeu *Sports*. Qu'en concluez vous ?

L'analyse de la variance permet d'étudier par exemple le comportement d'une variable qualitative à expliquer en fonction d'une ou de plusieurs variables nominales catégorielles. Cependant, certain tests sont applicables uniquement si les données suivent une loi normale. Il existe des tests statistiques permettant de savoir si une distribution suit la loi normale.

1.3 Un modèle dit "gaussien"

En théorie des probabilités et en statistique, la loi normale est l'une des lois de probabilités les plus adaptées pour modéliser des phénomènes naturels issus de plusieurs événements aléatoires. Elle est en lien avec de nombreux objets mathématiques dont le mouvement brownien, le bruit blanc gaussien pour ne citez qu'eux. Elle est également appelée loi gaussienne, loi de Gauss ou loi de Laplace-Gauss des noms de Laplace (1749-1827) et Gauss (1777-1855), deux mathématiciens, astronomes et physiciens qui l'ont étudiée.
Plus formellement, c'est une loi de probabilités absolument continue qui dépend de deux paramètres : son espérance, un nombre réel noté μ, et son écart type, un nombre réel strictement positif noté σ. La densité de probabilité de la loi normale est donnée par :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}(\frac{x-\mu}{\sigma})^2\right). \quad (1)$$

La courbe de cette densité est appelée **courbe de Gauss** ou **courbe en cloche**, entre autres. C'est la représentation la plus connue de cette loi. La loi normale

d'espérance nulle et d'écart type unitaire est appelée loi normale centrée réduite ou loi normale standard.
Lorsqu'une variable aléatoire X suit la loi normale, elle est dite gaussienne ou normale et il est habituel d'utiliser la notation avec la variance σ^2. Vous comprenez peut-être maintenant pourquoi nous vous avons obligé à connaître la fonction exponentielle. C'est grâce à cette fonction qu'on modélise cette forme de cloche représentative de la gaussienne. Nous allons essayer de comprendre le pic important suite à notre fonction **density()**.
En statistique, le test de Shapiro–Wilk teste l'hypothèse nulle (aussi appelé hypothèse H_o) selon laquelle un échantillon analysé est issu d'une population normalement distribuée. Nous allons regarder si les ventes sont normalement distribuée pour différents genre

```
> shapiro_test = stats.shapiro(df["Global_Sales"][df["Genre"]== "
    Adventure"])
print(shapiro_test)
ShapiroResult(statistic=0.3016361594200134, pvalue=0.0)
```

Travail à réaliser

a) Interprétez la p-value du test de Shapiro sur les jeux d'aventures

b) Interprétez la p-value du test de Shapiro sur les jeux de stratégie

c) Les ventes de *Aventure* et *stratégie* suivent-ils la loi normale ?

d) Testez la normalité sur la valeurs *Globale_Sales* de tout les genres de jeux. Que constatez vous ?

e) Pour faire un test de Shapiro, la taille de l'échantillon doit être comprise entre 3 et 5000 Réalisez un test plus adapté pour tester la distribution normal de *Global_Sales*. Quelle valeur trouvez vous pour la p-value [2] ?

f) Concluez.

1.4 Time to decide

La fonction `pointplot()` vous permet d'avoir la moyenne et l'intervalle de confiance. L'intervalle de confiance permet d'évaluer la précision de l'estimation d'un paramètre statistique sur un échantillon.

```
> ax = sns.pointplot(x=df["Genre"] ,  y=df["Global_Sales"], data=df
    , estimator=np.mean,capsize=.2)
> plt.figure()
> plt.show()
```

2. Vous êtes bloqué ? Le nom du test ressemble à une célèbre Vodka

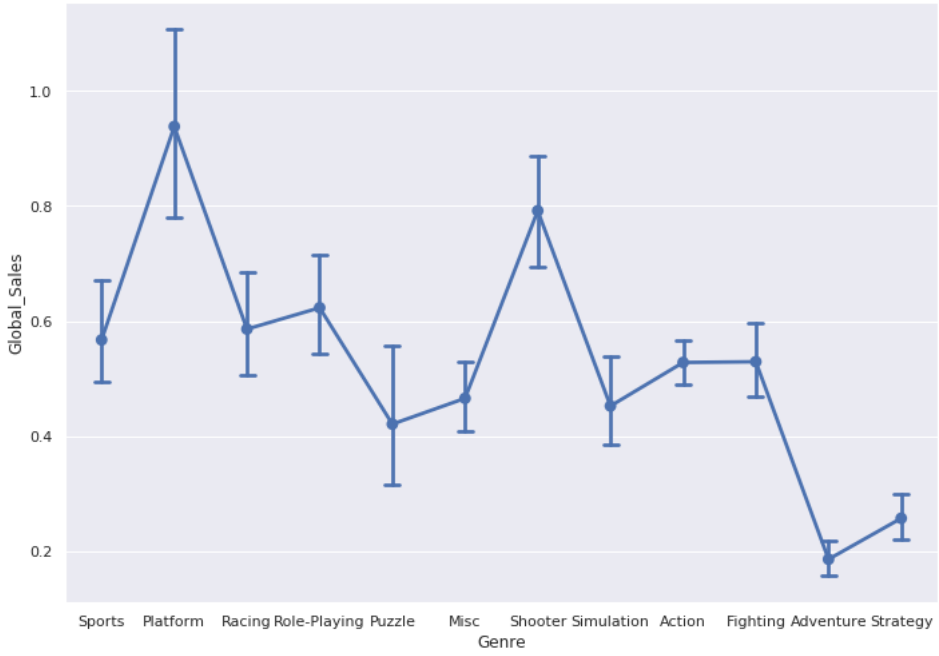

FIGURE 6 – `pointplot()` sur *Global_Sales* (IC à 95%)

Travail à réaliser

a) Citez la formule mathématique permettant de retrouver les valeurs définissant l'intervalle de confiance. Interprétez le graphique 6.

b) Quel genre n'a pas été bien venu ?

c) Quel genre a été le mieux venu ?

d) Quel pouvez vous dire sur le jeux qui à été le mieux vendu ? Pensez vous que ce jeux à vraiment été très populaire ?

e) Créez un sous ensemble de données, ayant uniquement la liste des jeux depuis 2014.

f) Affichez la fonction plotmeans des Global_Sales selon les genres pour ce nouveau datafame.

g) Quel genre n'a pas été bien venu, depuis 2014 ?

h) Quel genre a été le mieux venu, depuis 2014 ?

i) Appliquez un centrage réduction sur vos données depuis 2014 et comparer les résultats.

j) Dans quel genre de jeux pensez-vous qu'il est opportun d'aller ?

k) Sur quelle plate forme allez-vous proposer votre jeu ?

l) Dans quel pays allez-vous lancer votre jeu ?

m) Concluez sur ce jeu de données