

Parcours de Data Analyst avec
OPENCLASSROOMS

Projet 4 : Analysez les ventes de votre entreprise

Cédric PAPIN

Mentor: Benjamin Marlé



15 novembre 2018

- ☆ **Présentation du projet**
- ☆ **Nettoyage du jeu de données**
- ☆ **Différentes analyses effectuées**
- ☆ **Court fichier README**

I. Présentation du projet

- ★ **Présentation du projet**
- ★ **Nettoyage du jeu de données**
- ★ **Différentes analyses effectuées**
- ★ **Court fichier README**

I. Présentation du projet

- Nouvellement embauché dans une chaîne de librairies en tant que data analyst
- celle-ci a plusieurs boutiques ainsi qu'un site de vente en ligne
- fonctionnant avec un algorithme de recommandation
- l'entreprise connaît un véritable succès
- le boss m'a demandé de faire une brève présentation



II. Nettoyage du jeu de données

- ★ Présentation du projet
- ★ **Nettoyage du jeu de données**
- ★ Différentes analyses effectuées
- ★ Court fichier README

DATASET

3 tables sous format csv issues de la BDD de l'entreprise :

- les ventes appelées « Transactions »
- la liste des clients
- la liste des produits

Et hop, nous allons nettoyer tout ça ...



Valeurs aberrantes

```
In [8]: 1 # on trie la variable 'client_id' avec la méthode .sort_values()  
2 customers.sort_values(by='client_id', ascending=False).head()
```

Out[8]:

	client_id	sex	birth
8494	ct_1	m	2001
2735	ct_0	f	2001
7358	c_999	m	1964
2145	c_998	m	2001
94	c_997	f	1994

- ★ après tri, apparition de valeurs 'ct'
- ★ hypothèse : tests ultérieurs
- ★ suppression avec un .loc[]
- ★ traitement appliqué sur chaque table

```
3 customers = customers.loc[~customers.client_id.str.startswith('ct'), :]  
4 customers.sort_values(by='client_id', ascending=False).head()
```

Out[9]:

	client_id	sex	birth
7358	c_999	m	1964
2145	c_998	m	2001
94	c_997	f	1994
2788	c_996	f	1970
7004	c_995	m	1955

Jointure



Rien ne se perd tout se transforme ...

Rien n'est perdu

	client_id	sex	birth	id_prod		date	session_id	price	categ
0	c_4410	f	1967	0_1455	2021-03-22 14:29:25.189266	s_9942	8.99	0	
1	c_4389	m	1984	0_1455	2021-07-09 11:16:18.579726	s_59967	8.99	0	
2	c_5019	f	1977	0_1455	2022-01-15 00:01:53.456196	s_149928	8.99	0	
3	c_7049	f	1987	0_1455	2021-03-04 14:01:38.698752	s_1637	8.99	0	
4	c_5110	f	1982	0_1455	2021-09-05 11:48:41.065009	s_85364	8.99	0	

Valeurs manquantes

- ★ la méthode `.info()` est utilisée pour détecter d'éventuelles valeurs manquantes
- ★ le tout est exporté sous format `.csv`

```
In [23]: 1 # test pour détecter valeurs manquantes
2 jointure.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 336713 entries, 0 to 336712
Data columns (total 8 columns):
client_id    336713 non-null object
sex          336713 non-null object
birth         336713 non-null int64
id_prod       336713 non-null object
date          336713 non-null object
session_id   336713 non-null object
price         336713 non-null float64
categ         336713 non-null int64
dtypes: float64(1), int64(2), object(5)
memory usage: 23.1+ MB
```

```
In [17]: 1 # nous n'avons donc aucune valeur manquante
```

```
In [18]: 1 # exportation de la table sous format .csv
2 jointure.to_csv("jointure.csv", index=False)
```

III. Différentes analyses effectuées

- ★ Présentation du projet
- ★ Nettoyage du jeu de données
- ★ **Différentes analyses effectuées**
- ★ Court fichier README

Import des librairies utiles

- pandas
- numpy
- scipy.stats
- statsmodels.api
- seaborn
- matplotlib

Indicateurs de tendance centrale et de dispersion

- ☆ travail sur la variable ‘price’ de la table jointure
- ☆ 1 seule ligne de commande
- ☆ package très utile d’indicateurs

```
In [53]: 1 # calcul de la moyenne et de l'écart-type sur la variable 'price' de la table jointure
          2 jointure['price'].describe()

Out[53]: count    336713.000000
          mean      17.215189
          std       17.855445
          min       0.620000
          25%      8.610000
          50%     13.900000
          75%     18.990000
          max      300.000000
          Name: price, dtype: float64
```

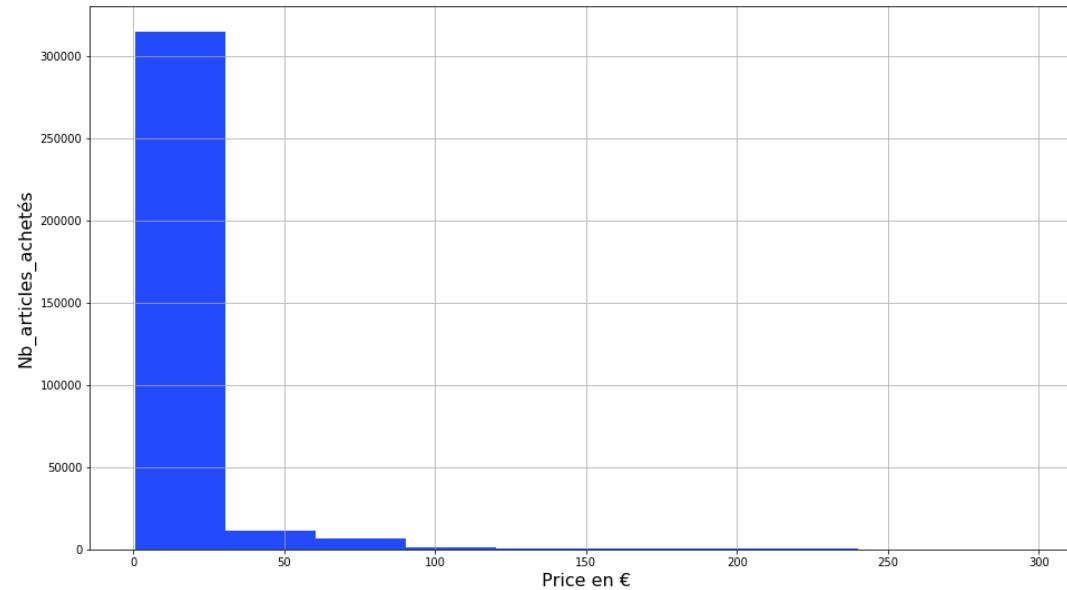
Histogramme

- sur une variable quantitative continue, les classes sont agrégées automatiquement
- différentes réglages possibles sur le graphique, dimension, couleur etc...
- légendes nommables et ajustables

```
In [55]: 1 # représentation graphique de la variable 'price' à l'aide d'un histogramme
2 jointure['price'].hist(figsize=(16,9), color='b')
3 plt.xlabel("Price en €", fontsize=16)
4 plt.ylabel("Nb_articles_achetés", fontsize=16)
5 plt.suptitle("Histogramme de la variable 'price'", fontsize=16)
```

```
Out[55]: Text(0.5,0.98,"Histogramme de la variable 'price'")
```

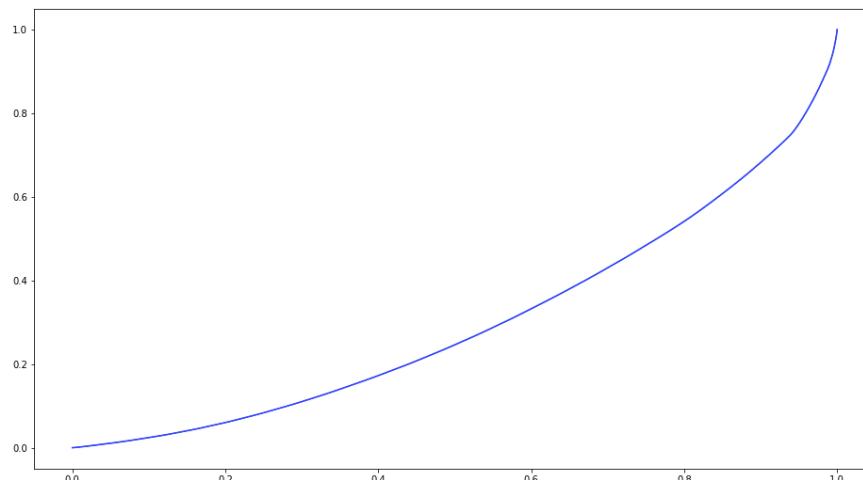
Histogramme de la variable 'price'



Analyse de concentration

Courbe de Lorenz

Courbe de Lorenz de la variable 'price'



Indice de Gini

```
In [57]: 1 # nous allons afficher l'indice de Gini sur la variable 'price'  
2 aire_ss_courbe = lorenz[:-1].sum()/len(lorenz) # aire sous la courbe de Lorenz. La dernière valeur ne participe pas  
3 S = 0.5 - aire_ss_courbe # aire entre la bissectrice et la courbe de Lorenz  
4 gini = 2*S  
5 gini
```

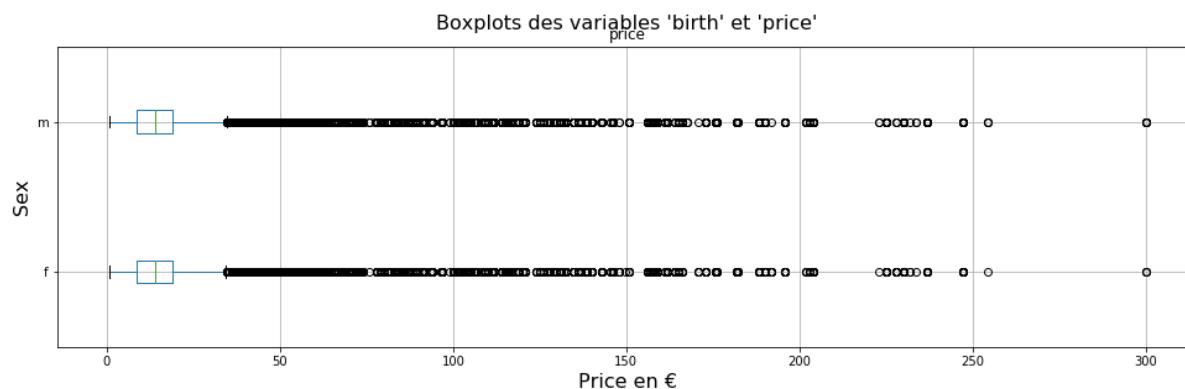
Out[57]: 0.39215028602493873

Boxplots

- ★ sur variables quali et quanti
- ★ très pratique pour faire une comparaison

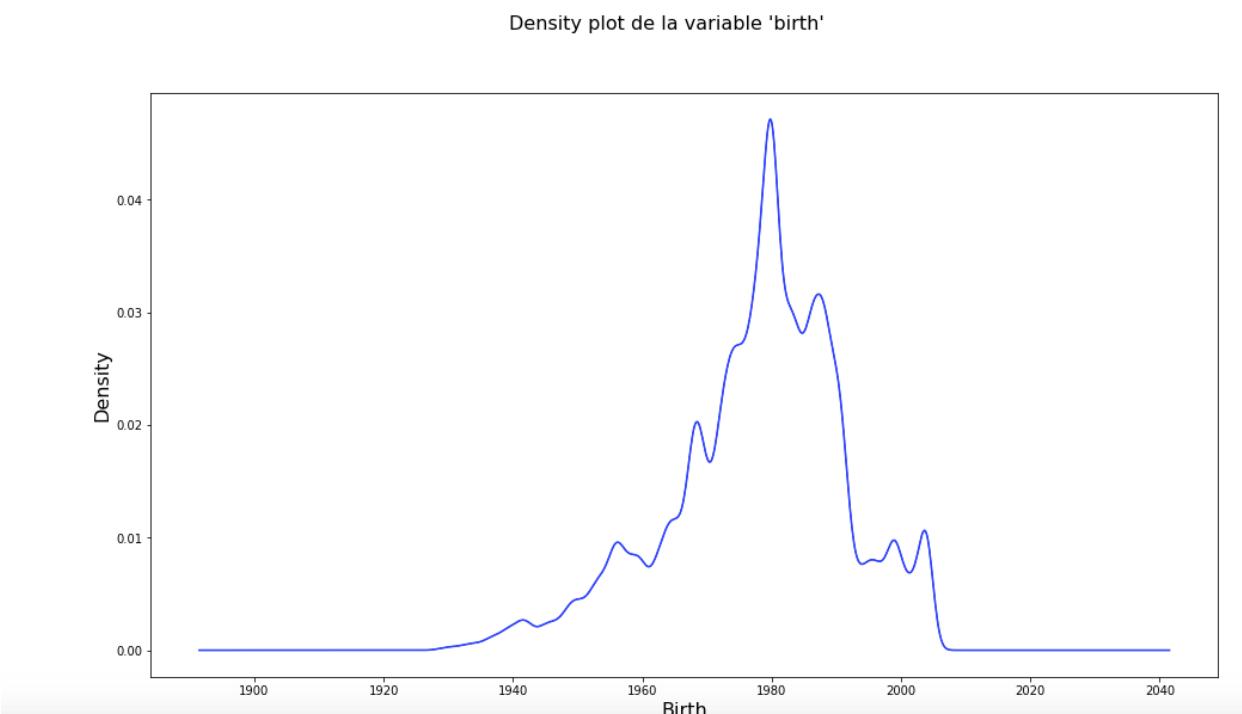
```
In [66]: 1 # nous allons afficher le boxplot prenant en variables :  
2 # qualitative : 'sex'  
3 # quantitative : 'price'  
4 jointure.boxplot(column='price',by='sex',vert=False,figsize=(16,4.5))  
5 plt.xlabel("Price en €", fontsize=16)  
6 plt.ylabel("Sex", fontsize=16)  
7 plt.suptitle("Boxplots des variables 'birth' et 'price'", fontsize=16)
```

Out[66]: Text(0.5,0.98,"Boxplots des variables 'birth' et 'price'")



Série temporelle

- l'axe des abscisses représente des dates
- le density plot est adapté pour ce type de représentation



Analyses bivariées

- ★ calcul du CA par client
- ★ présence d'outliers
- ★ création d'un sous-échantillon pour les isoler

```
In [11]: 1 # montant total des achats par client
2 gb_client_id = jointure.groupby('client_id')
3 gb_client_id = gb_client_id['price'].sum(min_count=1)
4 gb_client_id = gb_client_id.reset_index()
5 gb_client_id.columns = ['client_id','montant_achats']
6 gb_client_id['%_du_CA'] = (gb_client_id.montant_achats / gb_client_id.montant_achats.sum()) * 100
7 gb_client_id.sort_values(by='montant_achats',ascending=False)[0:8]
```

Out[11]:

	client_id	montant_achats	%_du_CA
677	c_1609	162007.34	2.794879
4388	c_4958	144257.21	2.488662
6337	c_6714	73197.34	1.262768
2724	c_3454	54442.92	0.939225
7715	c_7959	2564.25	0.044237
3870	c_4491	2540.53	0.043828
7791	c_8026	2537.67	0.043779
1268	c_2140	2527.01	0.043595

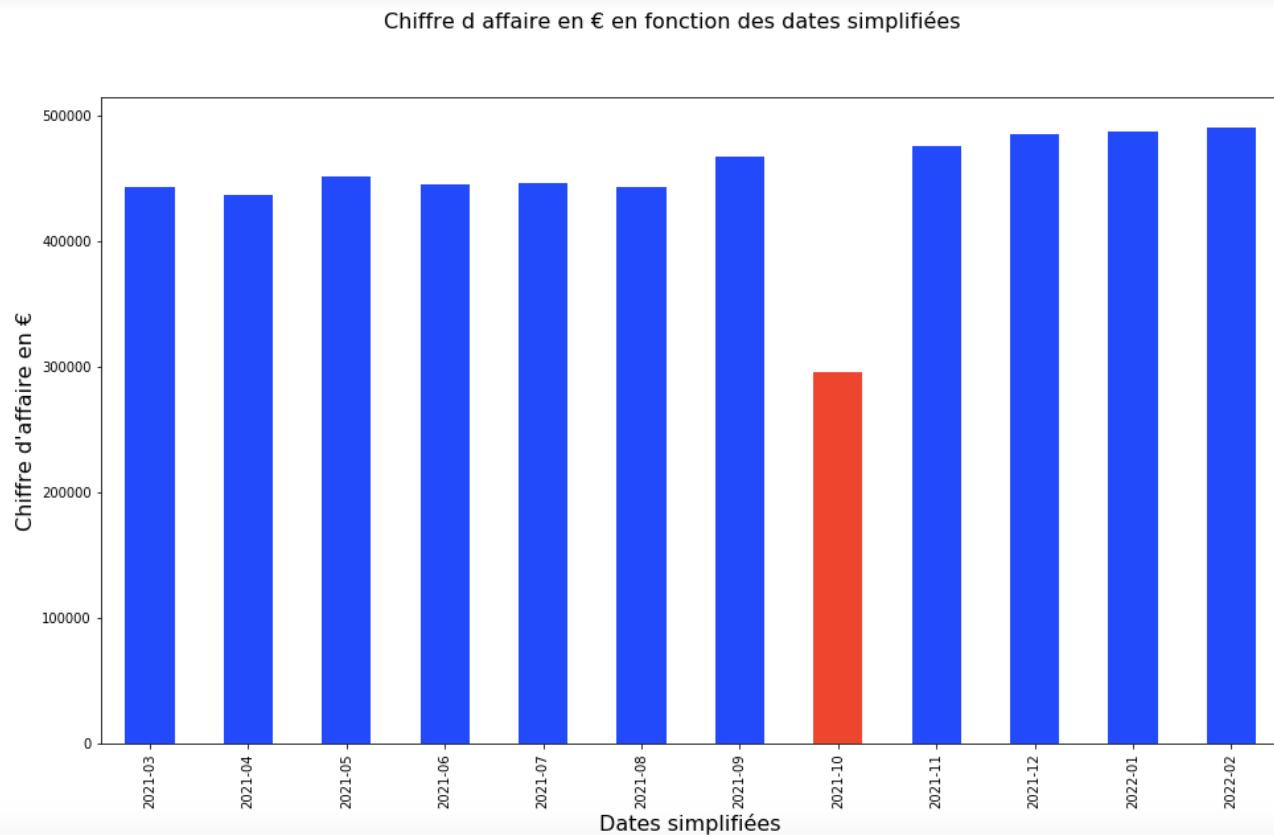
Variable 'date'

- transformation d'une variable 'str' en 'timestamps'
- création d'une nouvelle variable

```
In [166]: 1 # test du type de la variable 'date'  
2 type(sousJointure.date[0])  
  
Out[166]: str  
  
In [4]: 1 # nous allons redéfinir la date comme une date et non une variable textuelle  
2 sousJointure['date'] = pd.to_datetime(sousJointure['date'], errors='coerce')  
3 type(sousJointure.date[0])  
  
Out[4]: pandas._libs.tslibs.timestamps.Timestamp  
  
In [5]: 1 # nous allons modifier le format de la variable 'date'  
2 sousJointure['dates_simplifiees'] = sousJointure.date.map(lambda x: x.strftime('%Y-%m'))  
3 sousJointure.head()  
  
Out[5]:  
   client_id  sex  birth  id_prod          date  session_id  price  categ  dates_simplifiees  
0  c_4410     f  1967  0_1455  2021-03-22 14:29:25.189266  s_9942    8.99    0  2021-03  
1  c_4389     m  1984  0_1455  2021-07-09 11:16:18.579726  s_59967    8.99    0  2021-07  
2  c_5019     f  1977  0_1455  2022-01-15 00:01:53.456196  s_149928   8.99    0  2022-01  
3  c_7049     f  1987  0_1455  2021-03-04 14:01:38.698752  s_1637    8.99    0  2021-03  
4  c_5110     f  1982  0_1455  2021-09-05 11:48:41.065009  s_85364    8.99    0  2021-09
```

CA = WARNING

- alerte sur le CA
- recherche des causes

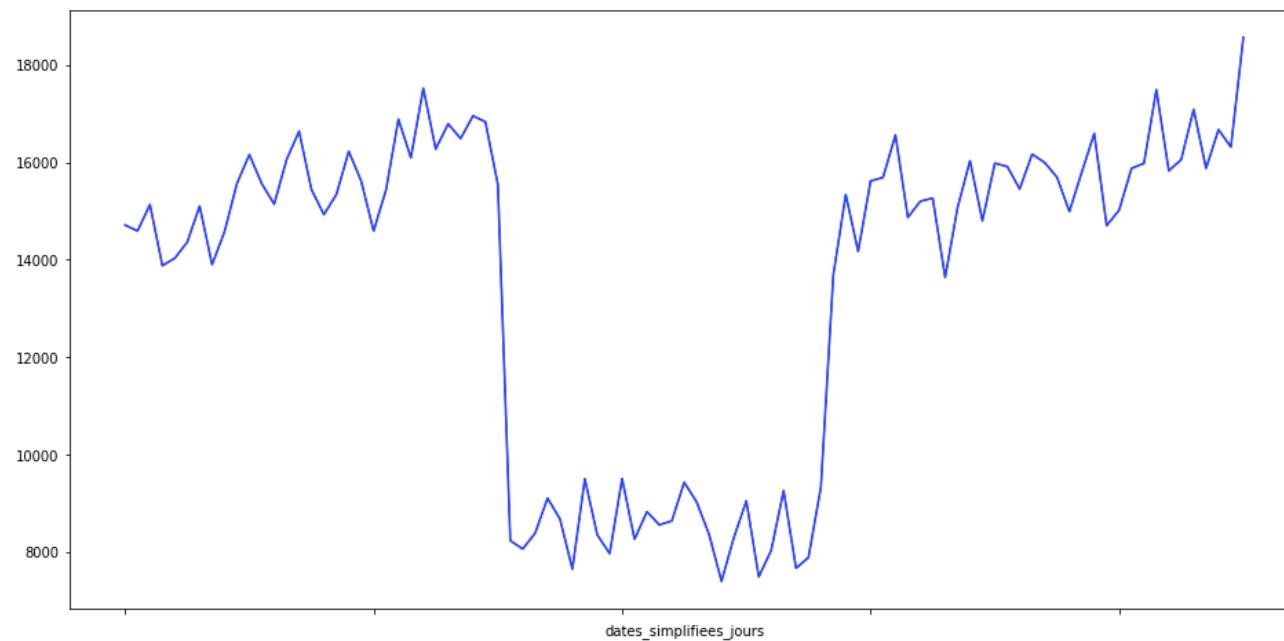


Recherches sur la table

- ★ affichage du plot issu des recherches
- ★ nous avons une baisse significative du CA en octobre 2021 sur 3 semaines

```
In [109]: 1 # affichage du plot
2 b['price'].plot(figsize=(16,8), color=['b'])

Out[109]: <matplotlib.axes._subplots.AxesSubplot at 0x1c26b7b7f0>
```



Corrélation entre sexe et categ ?

- travail sur 2 variables quali
- nous faisons appel à un tableau de contingence

categ	0	1	2	Total
sex				
f	101148.0	53774.0	8122.0	163044.0
m	94023.0	48851.0	7634.0	150508.0
total	195171.0	102625.0	15756.0	313552.0

- puis nous lançons un test d'indépendance

```
In [115]: 1 # pour vérifier ses observations, nous allons lancer un test d'indépendance
2 chi2, pvalue, degrees, expected = chi2_contingency(cont.iloc[0:2,0:3])
3 chi2, degrees, pvalue
4
```

```
Out[115]: (10.202417277273174, 2, 0.0060893822533516695)
```

```
In [116]: 1 # chi2 théorique = 5,99, celui-ci est inférieur à notre chi2 observé, donc dépendance entre les variables
2 # de plus pvalue inférieure à 0.05, les variables sont dépendantes donc sont corrélées
3 # cf pages 7/8 du doc
```

Interprétation des résultats

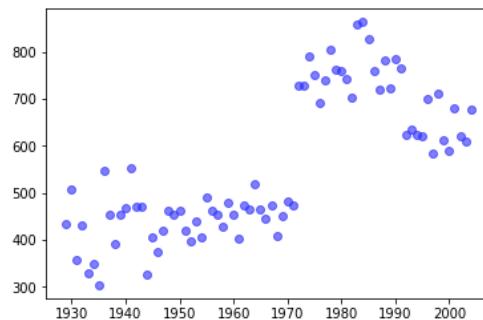
Les variables étudiées sont corrélées statistiquement mais cela n'implique pas forcément de lien de cause à effet (juste une forte chance).

Corrélation entre âge et montant ?

★ 2 variables quanti

```
In [173]: 1 # Affichage du scatter plot  
2 plt.plot(gb_age_total_achats['birth'],gb_age_total_achats["price"],'o',alpha=0.5, color='b')  
  
Out[173]: <matplotlib.lines.Line2D at 0x1c138988d0>
```

★ affichage du scatter plot



★ calcul du coefficient de détermination noté R2

```
In [24]: 1 # Calcul du coefficient de Pearson  
2 a = st.pearsonr(gb_age_total_achats["birth"],gb_age_total_achats["price"])[0]  
3 a  
  
Out[24]: 0.7489131423872507
```

```
In [122]: 1 # résultat identique avec une fonction d'équation  
2 gb_age_total_achats['birth'].corr(gb_age_total_achats["price"])  
  
Out[122]: 0.7489131423872508
```

```
In [25]: 1 # Calcul du coefficient de détermination noté R2  
2 R2 = a*a  
3 R2  
  
Out[25]: 0.5608708948403465
```

Interprétation des résultats

Les variables étudiées sont corrélées statistiquement mais cela n'implique pas forcément de lien de cause à effet (juste une forte chance).

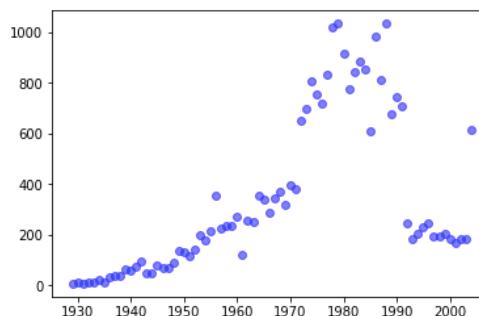
Corrélation entre âge et fréquence d'achat ?

★ 2 variables quanti

★ affichage du scatter plot

```
In [139]: 1 # Affichage du scatter plot
2 plt.plot(gb_frequence_achat['birth'],gb_frequence_achat["nb_achats_mois"],'o',alpha=0.5, color='b')

Out[139]: [<matplotlib.lines.Line2D at 0x1c265fc630>]
```



★ calcul du coefficient de détermination noté R2

```
]# Calcul du coefficient de Pearson
2 a = st.pearsonr(gb_frequence_achat["birth"],gb_frequence_achat["nb_achats_mois"])[0]
3 a
]0.5704319883233265

]# résultat identique avec une fonction d'équation
2 gb_frequence_achat['birth'].corr(gb_frequence_achat["nb_achats_mois"])
]0.5704319883233269

]# Calcul du coefficient de détermination noté R2
2 R2 = a*a
3 R2
]0.3253926533025037
```

Interprétation des résultats

Les variables étudiées sont corrélées statistiquement mais cela n'implique pas forcément de lien de cause à effet (juste une forte chance).

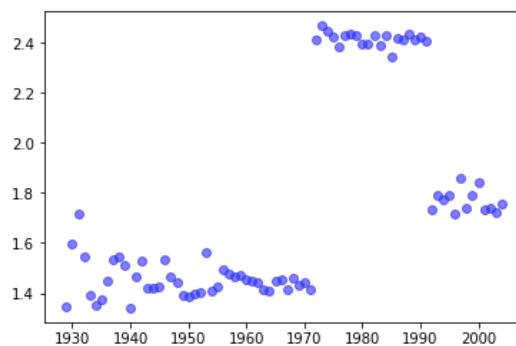
Corrélation entre âge et taille du panier moyen ?

★ 2 variables quanti

★ affichage du scatter plot

```
1 # Affichage du scatter plot
2 plt.plot(gb_panier_moyen['birth'],gb_panier_moyen['nb_articles'],'o',alpha=0.5, color='b')
```

[<matplotlib.lines.Line2D at 0x1c264c1c50>]



★ calcul du coefficient de

détermination noté R2

```
1 # Calcul du coefficient de Pearson
2 a = st.pearsonr(gb_panier_moyen["birth"],gb_panier_moyen["nb_articles"])[0]
3 a
```

0.5945228620310373

```
1 # résultat identique avec une fonction d'équation
2 gb_panier_moyen["birth"].corr(gb_panier_moyen["nb_articles"])
```

0.5945228620310374

```
1 # Calcul du coefficient de détermination noté R2
2 R2 = a*a
3 R2
```

0.35345743347757586

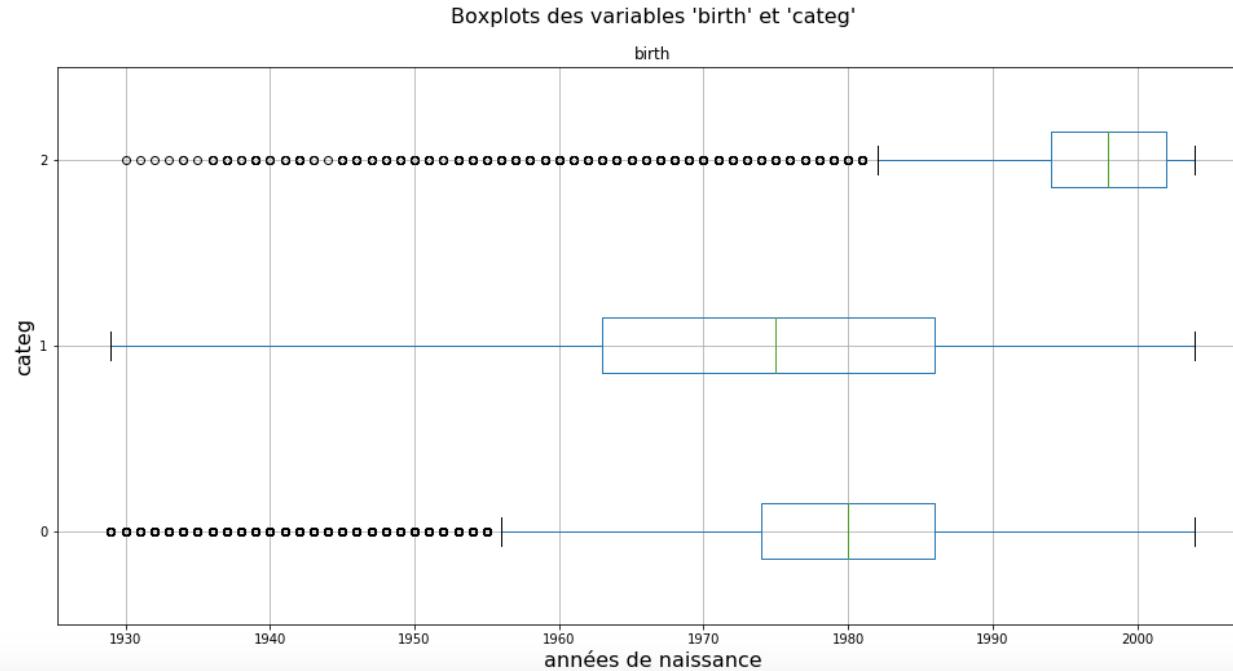
Corrélation entre âge et catégorie de produits achetés ?

★ 1 variable quanti et 1 variable quali

★ affichage de box plot

```
4 sousJointure.boxplot(column='birth', by='categ', vert=False, figsize=(16,8))
5 plt.xlabel("années de naissance", fontsize=16)
6 plt.ylabel("categ", fontsize=16)
7 plt.suptitle("Boxplots des variables 'birth' et 'categ'", fontsize=16)
```

55]: Text(0.5,0.98,"Boxplots des variables 'birth' et 'categ'")



Interprétation des résultats

Les variables étudiées sont corrélées statistiquement mais cela n'implique pas forcément de lien de cause à effet (juste une forte chance).

Calcul de η^2 (eta-squared)

```
6]: 1 # Calcul de  $\eta^2$  entre les variables 'birth' et 'categ'
2 X = "birth" # quantitative
3 Y = "categ" # qualitative
4
5 def eta_squared(x,y):
6     moyenne_y = y.mean()
7     classes = []
8     for classe in x.unique():
9         yi_classe = y[x==classe]
10        classes.append({'ni': len(yi_classe),
11                         'moyenne_classe': yi_classe.mean()})
12    SCT = sum([(yj-moyenne_y)**2 for yj in y])
13    SCE = sum([c['ni']*(c['moyenne_classe']-moyenne_y)**2 for c in classes])
14    return SCE/SCT
15
16 eta_squared(sous_jointure[X],sous_jointure[Y])
```

6]: 0.2667211293269678

Interprétation des résultats

Les variables étudiées sont corrélées statistiquement mais cela n'implique pas forcément de lien de cause à effet (juste une forte chance).

IV. Court fichier README

- ★ Présentation du projet
- ★ Nettoyage du jeu de données
- ★ Différentes analyses effectuées
- ★ Court fichier README

Court fichier README

Court fichier sous format .txt contenant des informations sur les autres livrables du projet

jupyter README.txt 12 hours ago

Logout

File Edit View Language Plain Text

```
1 Script 'Mission n°1 nettoyage'
2
3 - import 3 bases : customers.csv, products.csv, transactions.csv
4 - traitement : suppression des valeurs correspondantes aux tests, test présence de doublon et valeur manquante
5 - jointure d'une table reprenant les 3 bases
6 - exportation de cette table
7
8
9 Script 'Mission n°2 analyse + Mission n°3 corrélations'
10
11 - import des librairies nécessaires : pandas, numpy, scipy.stats, statsmodels.api, seaborn et matplotlib
12 - analyses bivariées sur variables utiles
13 - édition de graphiques
14 - interprétations des calculs sur les possibles corrélations entre variables
15
16
17 Script 'Graphiques'
18
19 - ensemble des graphiques sous format .png
20
21
22
```

Remerciements

- OPENCLASSROOMS
- Benjamin Marlé, mon mentor
- Le mentor en charge de ma soutenance pour ce projet

