

Projet 4 : Analyser les ventes de votre entreprise

Année 2018-2019





Analyser les ventes de ma
nouvelle entreprise



I-Nettoyage des données

3 jeux de données

client

	client_id	sex	birth
0	c_4410	f	1967
1	c_7839	f	1975
2	c_1699	f	1984

produit

	id_prod	price	categ
0	0_1421	19.99	0
1	0_1368	5.13	0
2	0_731	17.99	0

Transaction

	id_prod	date	session_id	client_id
0	0_1483	2021-04-10 18:37:28.723910	s_18746	c_4450
1	2_226	2022-02-03 01:55:53.276402	s_159142	c_277
2	1_374	2021-09-23 15:13:46.938559	s_94290	c_4270

Produit :

```
produit.describe() #Ici on observe qu'il y a un price négatif chose impossible
```

	price
count	3287.000000
mean	21.856641
std	29.847908
min	-1.000000
25%	6.990000
50%	13.060000
75%	22.990000
max	300.000000

```
print(produit.loc[produit['price'] < 0, :])#On va identifier ce prix négatif rapidement
```

	id_prod	price	categ
731	T_0	-1.0	0

Transaction :

	id_prod		date	session_id	client_id
0	0_1483	2021-04-10 18:37:28.723910		s_18746	c_4450
1	2_226	2022-02-03 01:55:53.276402		s_159142	c_277
2	1_374	2021-09-23 15:13:46.938559		s_94290	c_4270

```
transaction= transaction[~transaction['id_prod'].str.startswith('T_')]#On nettoie du coup p
```

Client :

	client_id	sex	birth
0	c_4410	f	1967
1	c_7839	f	1975
2	c_1699	f	1984

```
client=client[~client['client_id'].str.startswith('ct_')]
```

Fusion des jeux de données

```
Merge1=transaction.merge(produit, left_on='id_prod', right_on='id_prod', how='left')  
#On réalise un merge à gauche pour vérifier qu'on ne perd pas des produits avec un inner
```

	id_prod		date	session_id	client_id
0	0_1483	2021-04-10 18:37:28.723910		s_18746	c_4450
1	2_226	2022-02-03 01:55:53.276402		s_159142	c_277
2	1_374	2021-09-23 15:13:46.938559		s_94290	c_4270

	id_prod	price	categ
0	0_1421	19.99	0
1	0_1368	5.13	0
2	0_731	17.99	0

Fusion des jeux de données

```
sell=Merge1.merge(client, left_on='client_id', right_on='client_id', how='left') #On continue pour tout fusionner
```

	id_prod	date	session_id	client_id	price	categ
0	0_1483	2021-04-10 18:37:28.723910	s_18746	c_4450	4.99	0
1	2_226	2022-02-03 01:55:53.276402	s_159142	c_277	65.75	2
2	1_374	2021-09-23 15:13:46.938559	s_94290	c_4270	10.71	1
3	0_2186	2021-10-17 03:27:18.783634	s_105936	c_4597	4.20	0

	client_id	sex	birth
0	c_4410	f	1967
1	c_7839	f	1975
2	c_1699	f	1984

Recherche d'erreurs

```
sell.info()  
#Ici on affiche notre jeu de données pour vérifier le nombre non null et observer des incohérences  
  
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 336816 entries, 0 to 336815  
Data columns (total 8 columns):  
id_prod      336816 non-null object  
date         336816 non-null datetime64[ns]  
session_id   336816 non-null object  
client_id    336816 non-null object  
price        336713 non-null float64  
categ        336713 non-null category  
sex          336816 non-null category  
birth        336816 non-null int64  
dtypes: category(2), datetime64[ns](1), float64(1), int64(1), object(3)  
memory usage: 28.6+ MB
```

103 valeurs nulles

Que faire avec les valeurs manquantes ?

Solution 1

Supprimer les lignes : si les NAN<5% du jeu de données

Solution 2

**Inputation par valeur fixe (on remplace en effectuant :
moyenne, median etc...)**

Solution 3

Inputation par méthode de prédiction, remplacer par ce qui aurait pu être

Solution 4

Si les nombres de NAN sont trop importants, et si les autres solutions ne marchent pas alors on les supprimera

Les deux solutions étaient viables : mais pour conserver un maximum de données

```
moyenne_cat0=sell.loc[sell['categ']=='0','price'].mean()
#Ici on s'aperçoit que le début de l'id production correspond en général à sa catégorie de produit.
#Ici on n'observe alors que nous avons que des produits de catégorie 0
#Ici nous faisons tout simplement la moyenne des produits de catégorie pour remplacer ensuite sur les NAN
```

```
print(moyenne_cat0)
```

```
10.646828235274288
```

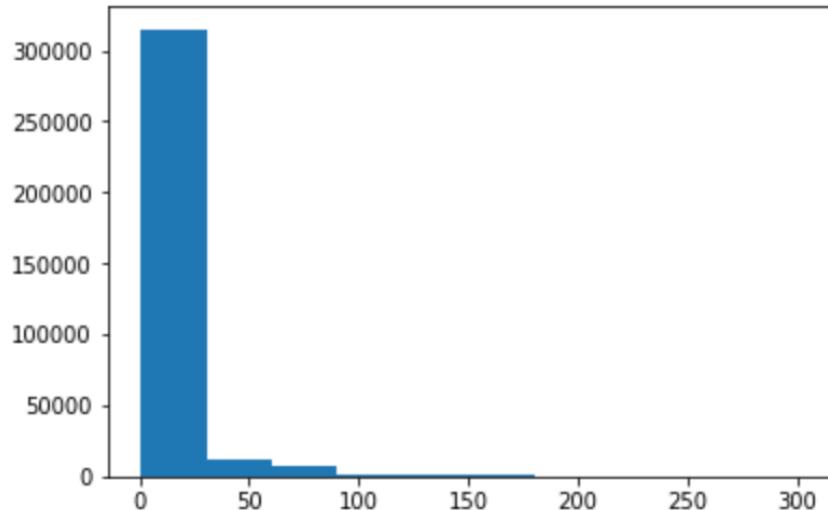
```
sell.loc[sell['price'].isnull(), 'price'] = 10.646828235274288
sell.loc[sell['categ'].isnull(), 'categ'] ='0'
#Nous avons donc calculé la moyenne d'un produit de catégorie 0 et nous remplaçons
#Et comme nous savons que se sont des produits de categorie 0, on va l'écrire
```

II-Analyse des données

Histogramme des prix des produits

```
plt.hist(sell['price'])

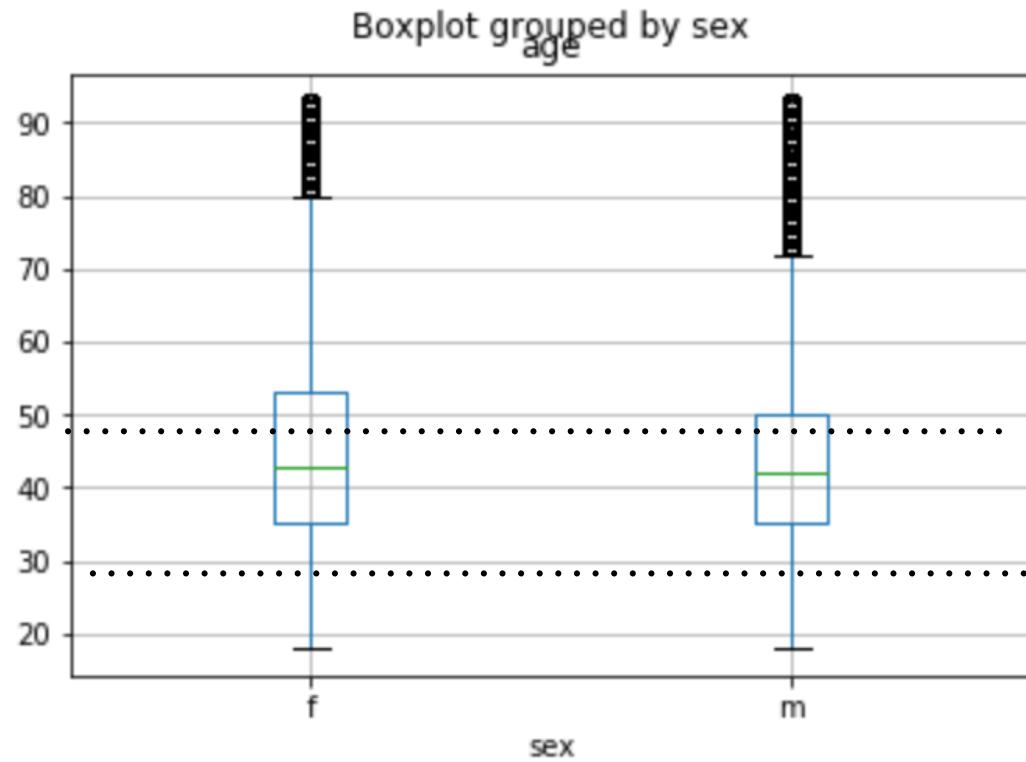
(array([3.14988e+05, 1.16870e+04, 6.83600e+03, 1.48300e+03, 5.20000e+02,
       6.47000e+02, 3.16000e+02, 2.61000e+02, 7.00000e+01, 8.00000e+00]),
 array([ 0.62 ,  30.558,  60.496,  90.434, 120.372, 150.31 , 180.248,
        210.186, 240.124, 270.062, 300.   ]),
 <a list of 10 Patch objects>)
```



Boîte à moustache des clients selon le sexe et l'âge

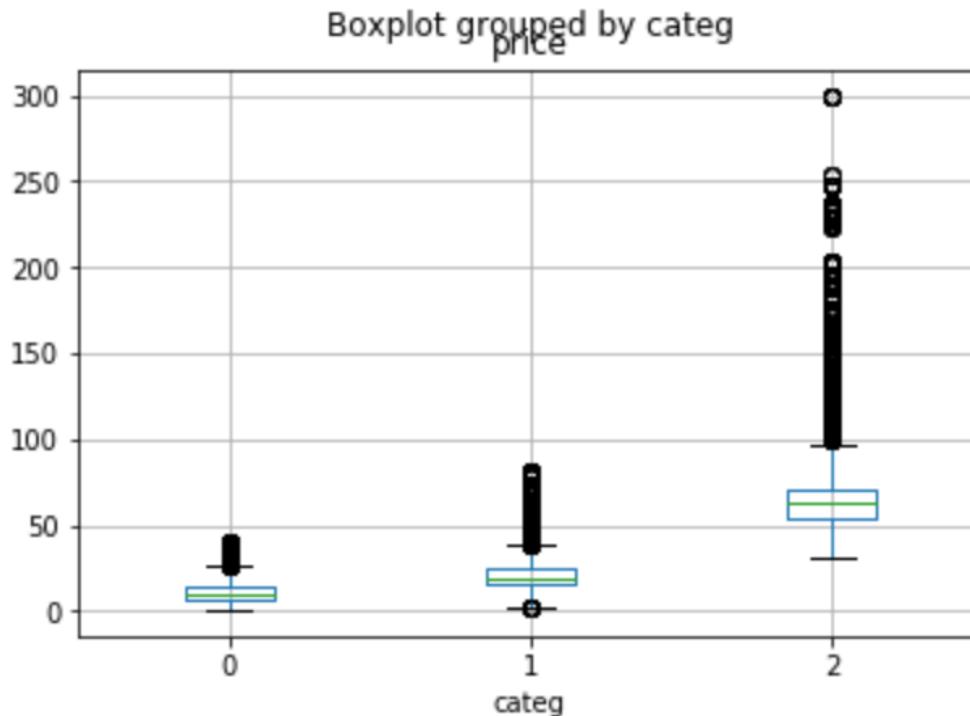
```
sell.boxplot(column='age', by='sex')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1aa51978>
```

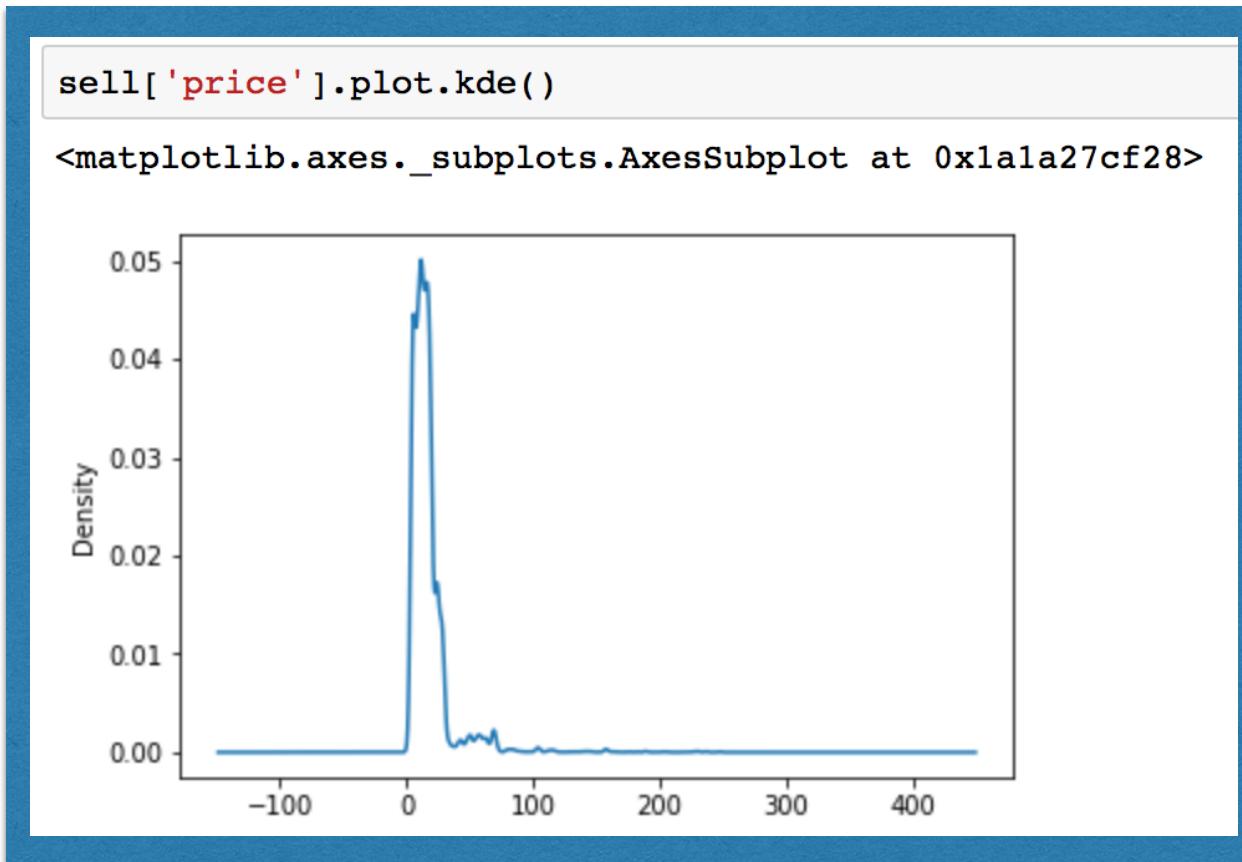


Boîte à moustache des produits selon le prix et leur catégorie

```
sell.boxplot(column='price', by='categ')  
<matplotlib.axes._subplots.AxesSubplot at 0x1a1a1bbf98>
```



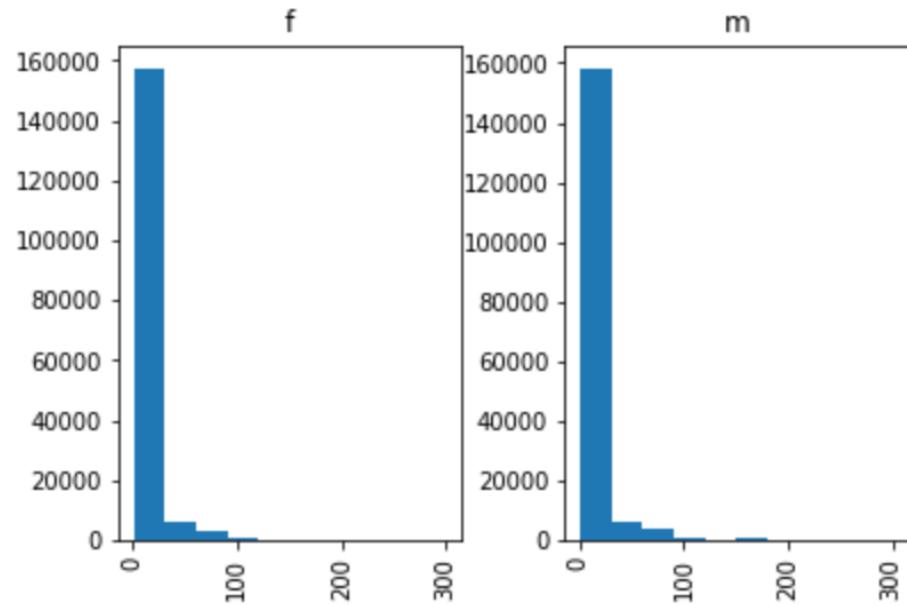
Densité des ventes en fonction du prix



Histogramme des ventes en fonction du prix et du sexe

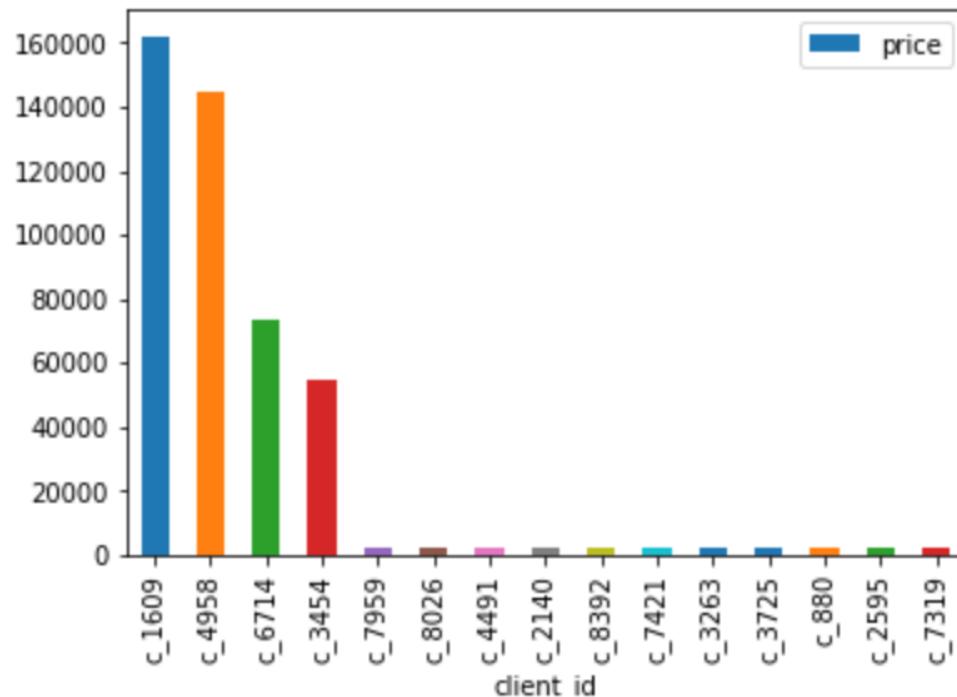
```
sell.hist(column='price', by='sex')

array([<matplotlib.axes._subplots.AxesSubplot object at 0x1a1bc25c50>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a1ae07400>],
      dtype=object)
```



Représentation des clients selon leur chiffre d'affaire

```
top4_client=['c_1609', "c_4958", "c_6714", "c_3454"]  
  
client.reset_index(inplace=True)  
client.head(n=15).plot.bar(x='client_id', y='price')  
  
<matplotlib.axes._subplots.AxesSubplot at 0x1a0f4d1550>
```

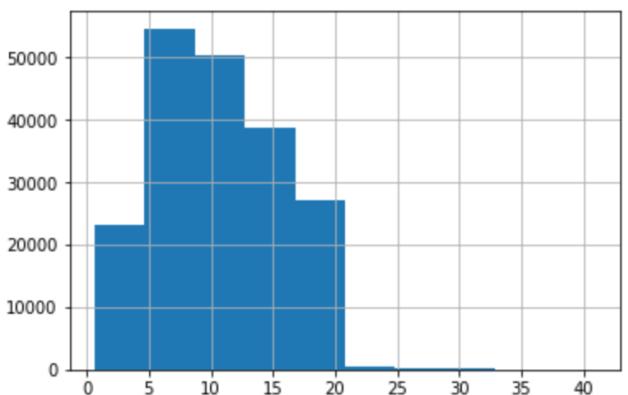


Analyse des catégories de prix

Catégorie 1

moyenne:
10.645050678519459

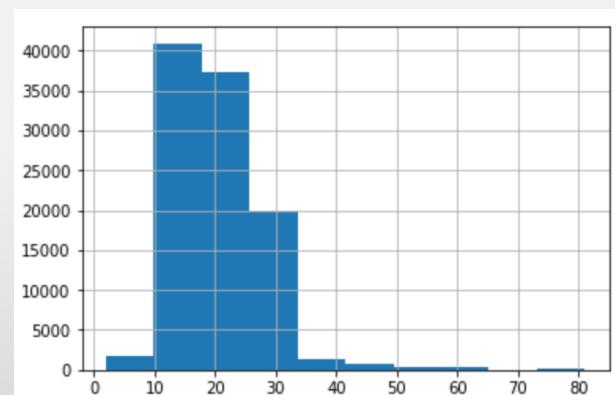
médiane:
9.99



Catégorie 2

moyenne:
20.478298075515564

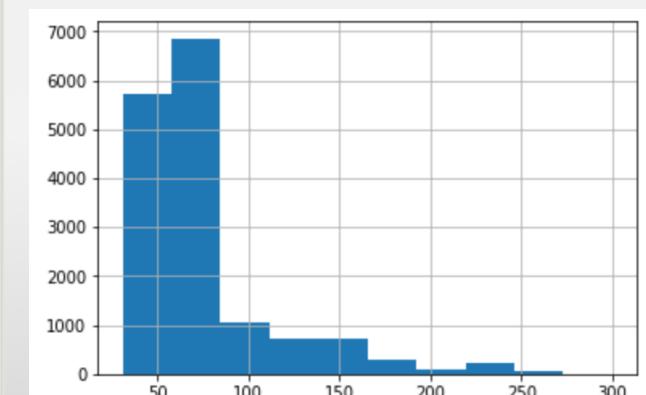
médiane:
19.08



Catégorie 3

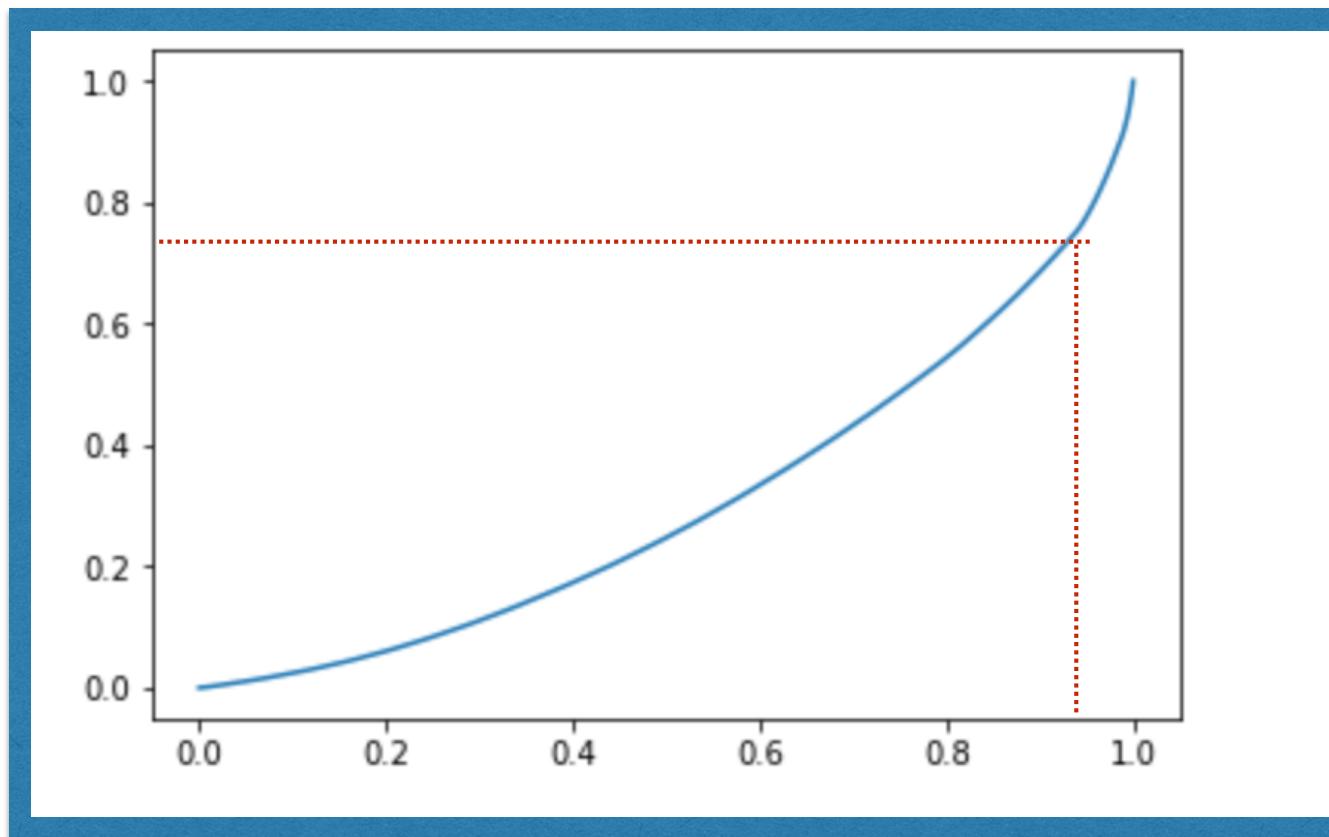
moyenne:
75.11313848692225

médiane:
62.54



Courbe de Lorentz

Chiffre
d'affaire

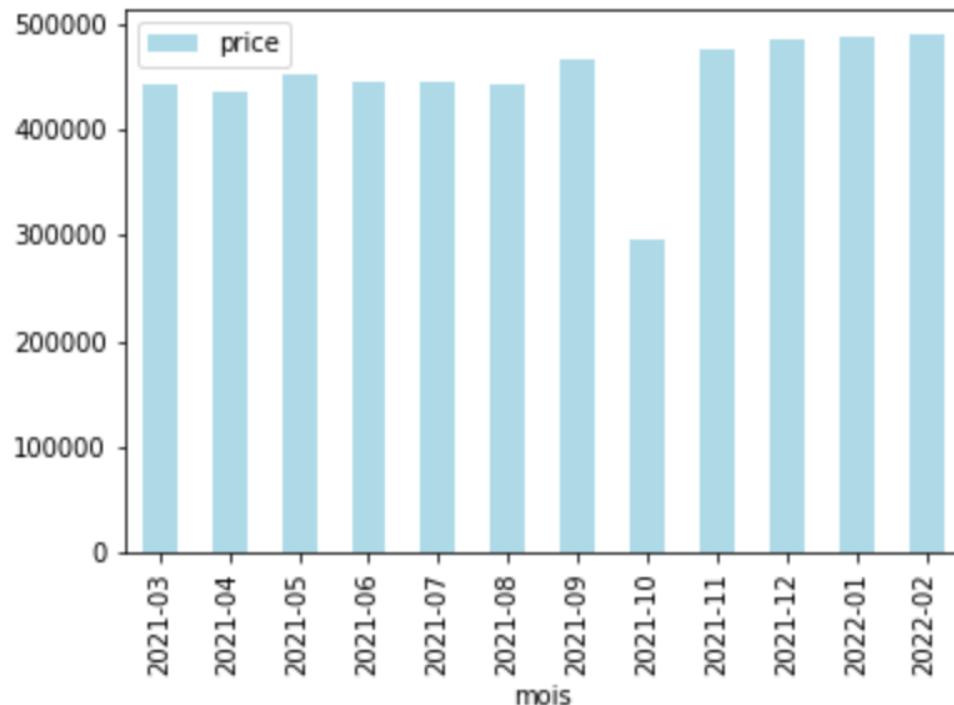


clients
Indice de Gini : 0.389

Détection d'un problème

```
time_serie.plot.bar(y='price',color='lightblue')  
#Ici on s'aperçoit que nos ventes ont baissé en octobre 2021
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a22b41630>
```



Investigation

```

Analyse=(sell.loc[(sell['date']>=pd.to_datetime('2021-10-01',format='%Y-%m-%d'))
                  &(sell['date']<=pd.to_datetime('2021-10-31',format='%Y-%m-%d')),:])
#On stocke notre mois suspect dans Analyse

Analyse_Nov=(sell.loc[(sell['date']>=pd.to_datetime('2021-11-01',format='%Y-%m-%d'))
                      &(sell['date']<=pd.to_datetime('2021-11-30',format='%Y-%m-%d')),:])
#On stocke un mois lambda pour comparer avec le mois d'octobre qui était suspect

```

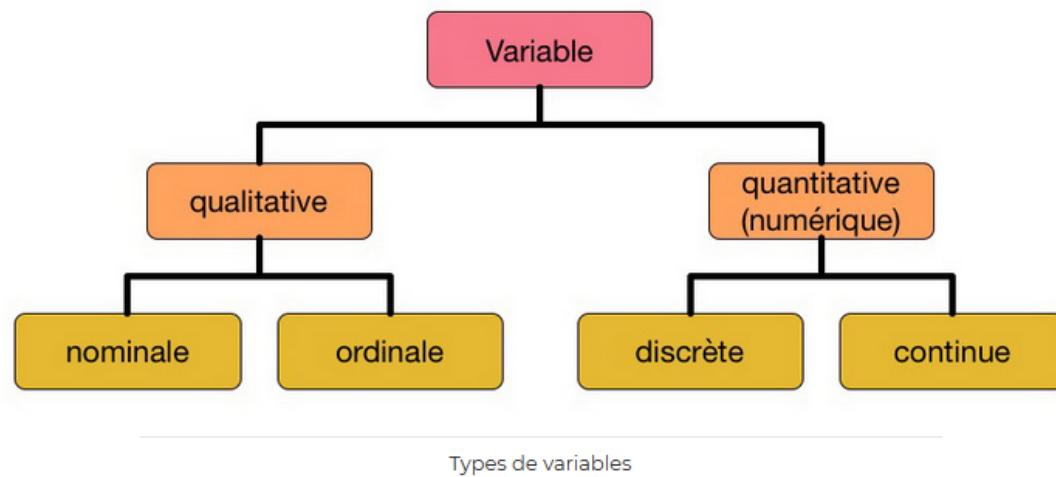
```

print(Analyse["categ"].value_counts())
print(Analyse_Nov['categ'].value_counts())
#Ici nous trouvons notre explication le mois d'octobre a vendu bcp plus de categorie 0 et 10 fois moins de categorie
#D'où la baisse de prix

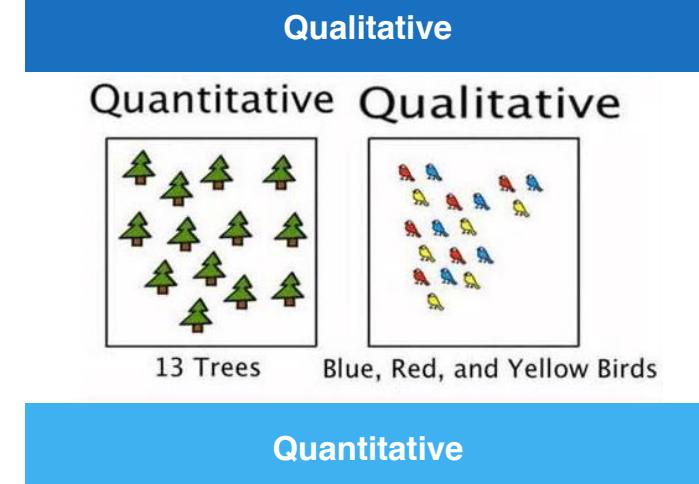
0    16879
1    1243
2    1021
Name: categ, dtype: int64
0    13160
1    11027
2    1189
Name: categ, dtype: int64

```

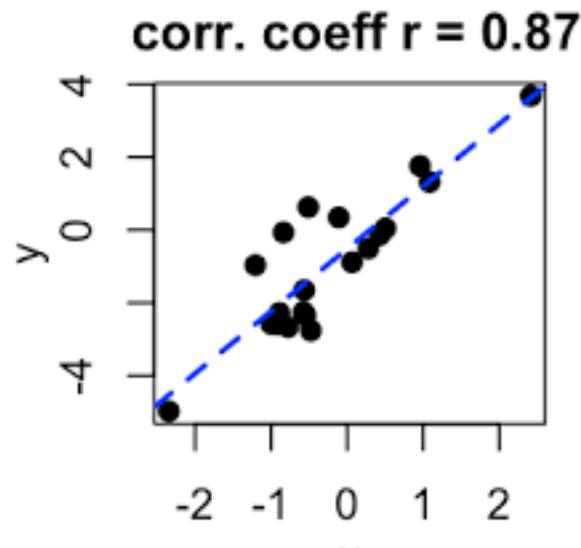
Type de variables



Type de variables



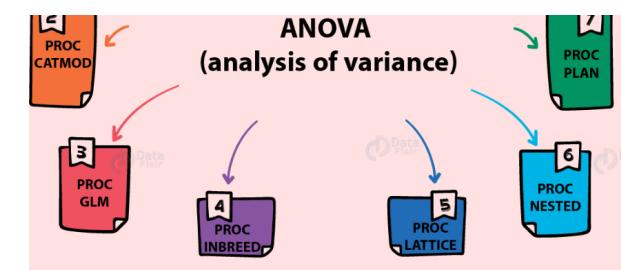
Les méthodes selon les types de données



Variables quantitatives

categ	0	1	2
sex			
f	101206	53774	8122
m	94064	48851	7634

Variables qualitatives



Variable qualitative et quantitative

Corrélation entre le sexe des clients et les catégories de produits achetés

categ	0	1	2
sex			
f	101206	53774	8122
m	94064	48851	7634

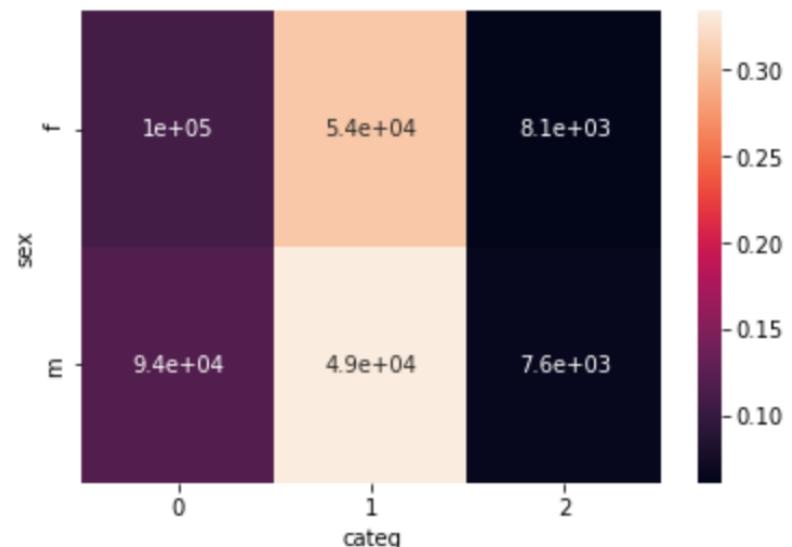
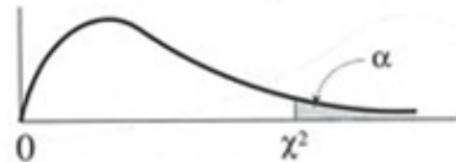


Tableau de contingence

Degré de liberté = nombre_lignes-1 X nombres_colonnes-1
Degré de liberté = 2

Heatmap

khi2 observé : 10.1108

Table χ^2 : points de pourcentage supérieurs de la distribution χ^2 

dl	.995	.990	.975	.950	.900	.750	.500	.250	.100	.050	.025	.010	.005
1	0.00	0.00	0.00	0.00	0.02	0.10	0.45	1.32	2.71	3.84	5.02	6.63	7.88
2	0.01	0.02	0.05	0.10	0.21	0.58	1.39	2.77	4.61	5.99	7.38	9.21	10.60
3	0.07	0.11	0.22	0.35	0.58	1.21	2.37	4.11	6.25	7.82	9.35	11.35	12.84
4	0.21	0.30	0.48	0.71	1.06	1.92	3.36	5.39	7.78	9.49	11.14	13.28	14.86
5	0.41	0.55	0.83	1.15	1.61	2.67	4.35	6.63	9.24	11.07	12.83	15.09	16.75
6	0.68	0.87	1.24	1.64	2.20	3.45	5.35	7.84	10.64	12.59	14.45	16.81	18.55
7	0.99	1.24	1.69	2.17	2.83	4.25	6.35	9.04	12.02	14.07	16.01	18.48	20.28
8	1.34	1.65	2.18	2.73	3.49	5.07	7.34	10.22	13.36	15.51	17.54	20.09	21.96
9	1.73	2.09	2.70	3.33	4.17	5.90	8.34	11.39	14.68	16.92	19.02	21.66	23.59
10	2.15	2.56	3.25	3.94	4.87	6.74	9.34	12.55	15.99	18.31	20.48	23.21	25.19
11	2.60	3.05	3.82	4.57	5.58	7.58	10.34	13.70	17.28	19.68	21.92	24.72	26.75
12	3.07	3.57	4.40	5.23	6.30	8.44	11.34	14.85	18.55	21.03	23.34	26.21	28.30

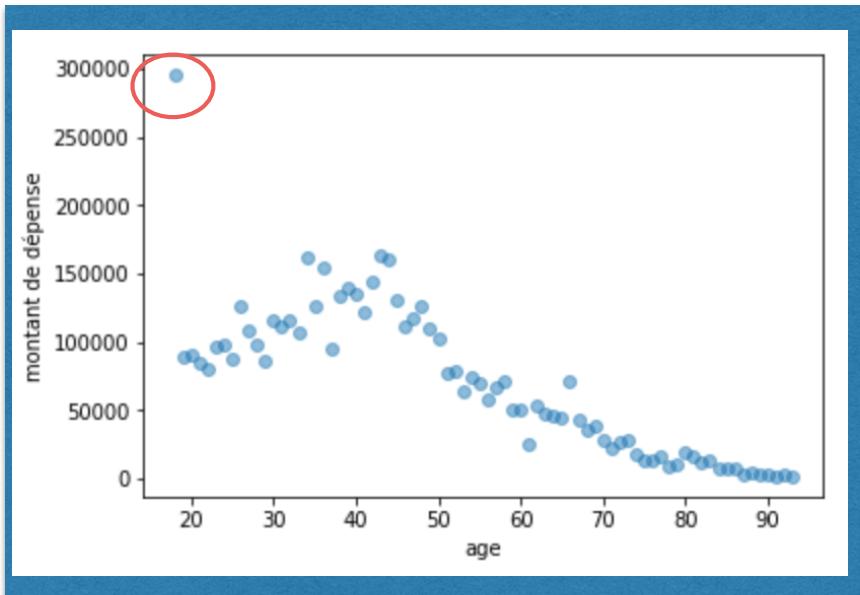
Si $\chi^2_{\text{observé}} > \chi^2_{\text{théorique}}$:

$$10,1108 > 5,99$$

on rejette l'hypothèse d'indépendance et il y a un lien de dépendance

Corrélation entre l'âge des clients et le montant total des achats

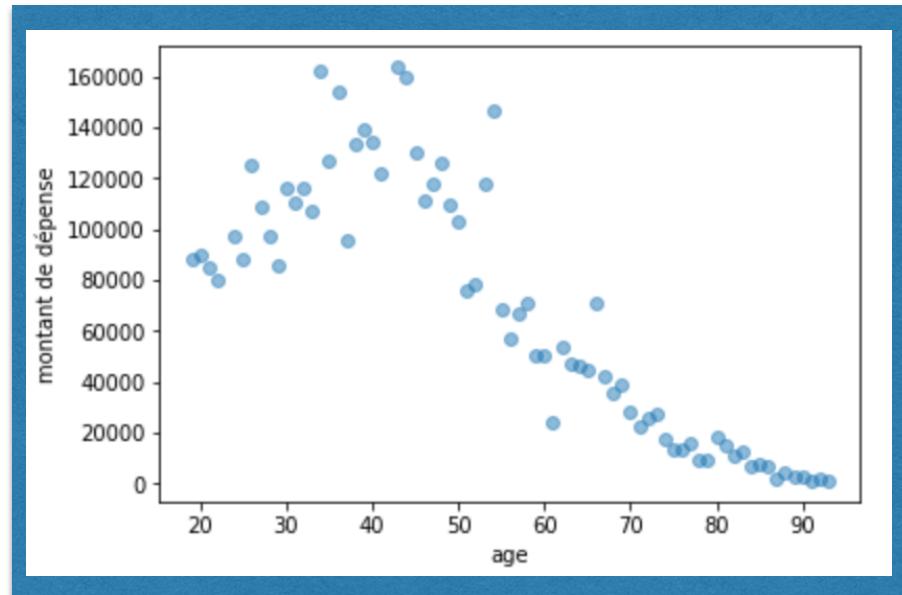
Avec Outliers



Pearson : -0.827

Covariance:-996419.93

Sans Outliers



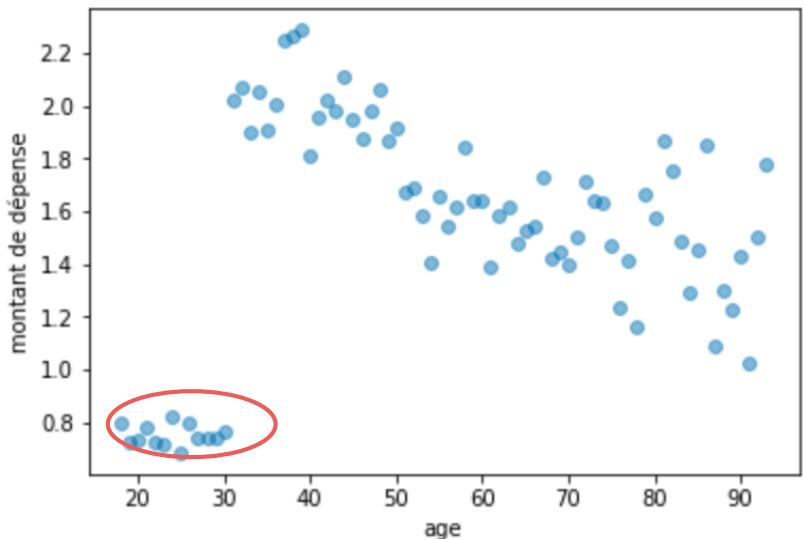
Pearson : -0.8397240

Covariance:-897033.552

Corrélation	Négative	Positive
Faible	de -0,5 à 0,0	de 0,0 à 0,5
Forte	de -1,0 à -0,5	de 0,5 à 1,0

Corrélation entre l'âge des clients et la fréquence d'achat

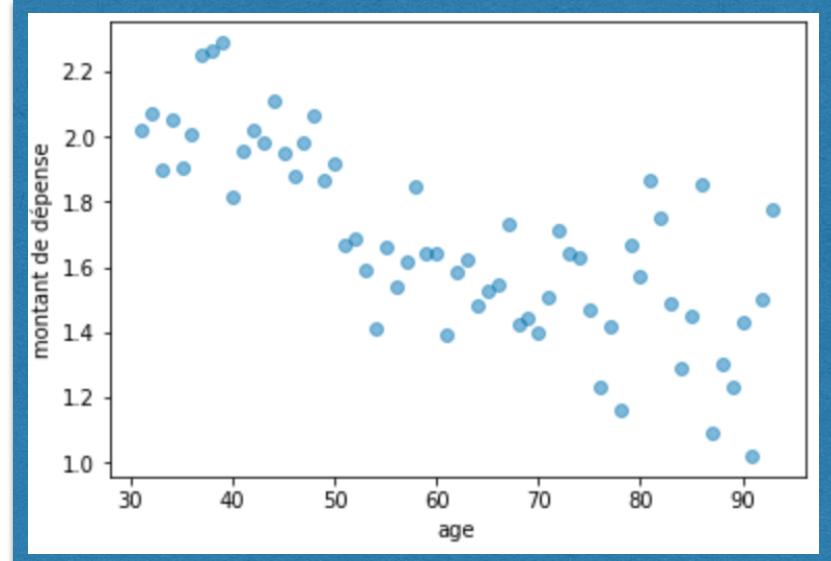
Avec Outliers



Pearson : -0.1777

Covariance:-1.70744

Sans Outliers



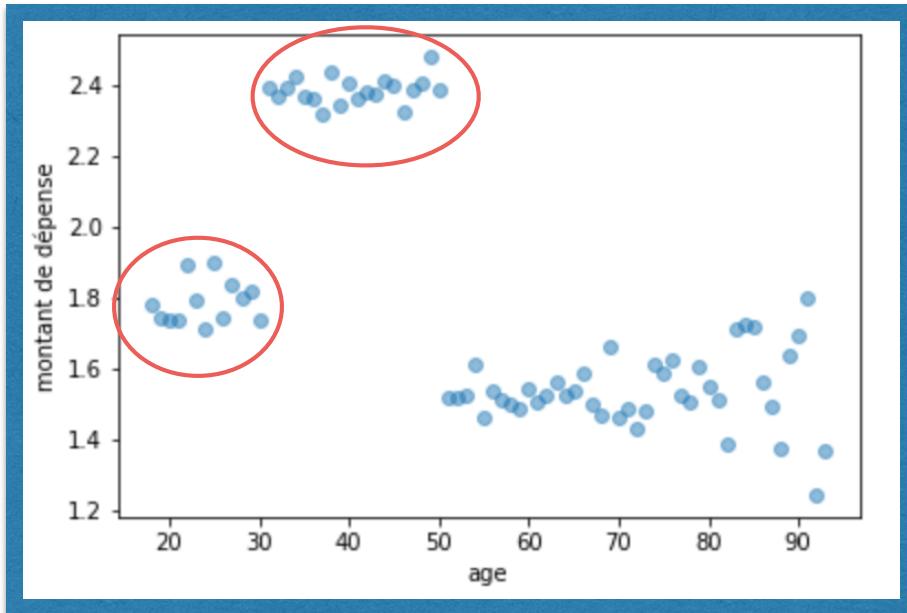
Pearson : -0.754

Covariance:-3.9708

Corrélation	Négative	Positive
Faible	de -0,5 à 0,0	de 0,0 à 0,5
Forte	de -1,0 à -0,5	de 0,5 à 1,0

Corrélation entre l'âge des clients et taille du panier moyen (en nombre d'articles)

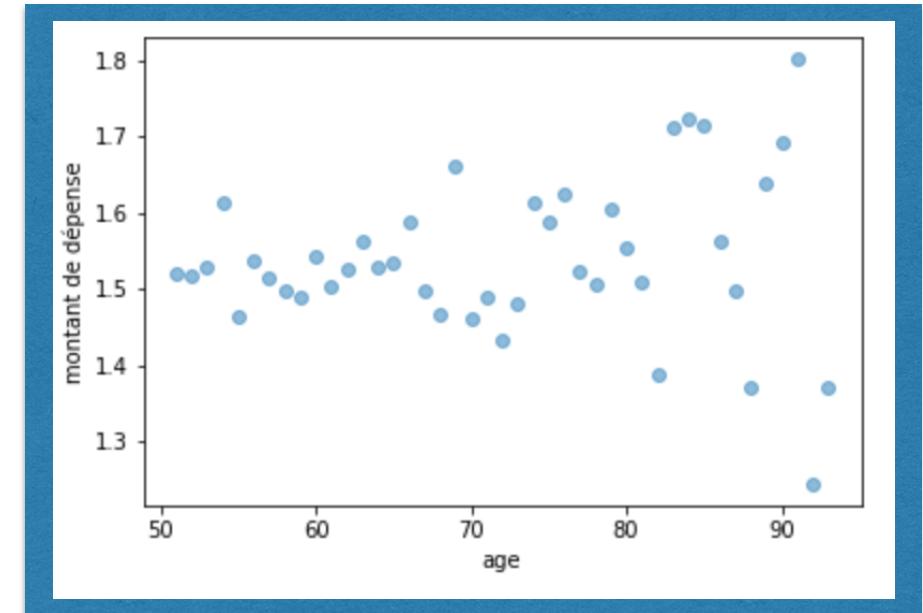
Avec Outliers



Pearson : 0.5679

Covariance: 4.596

Sans Outliers



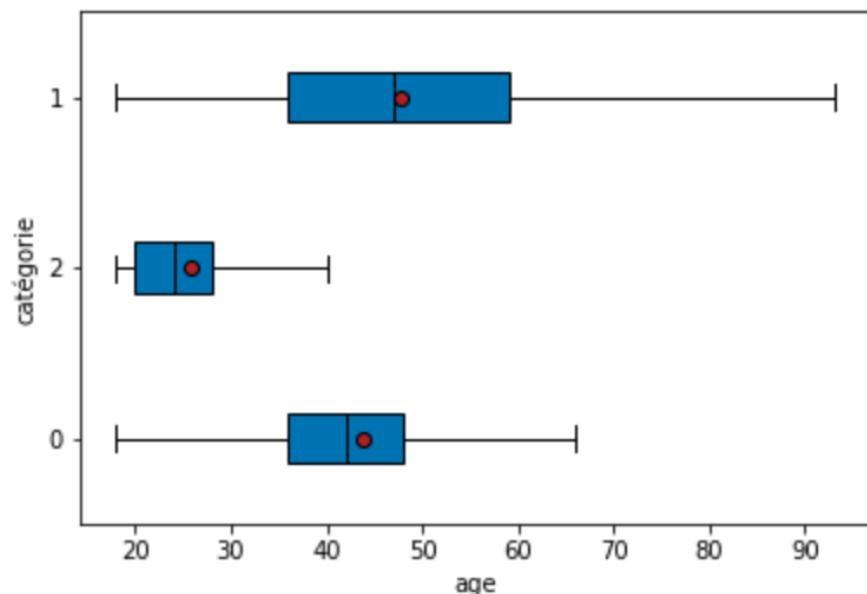
Pearson : -0.10163

Covariance: -0.12969

Corrélation	Négative	Positive
Faible	de -0,5 à 0,0	de 0,0 à 0,5
Forte	de -1,0 à -0,5	de 0,5 à 1,0

Eventuel corrélation avec catégorie d'âge

Corrélation entre l'âge des clients et les catégories de produits acheté



ANOVA

Eta carré : eta squared

Interpret η^2 as for r^2 or R^2 ; a rule of thumb (Cohen):

- .02 ~ small
- .13 ~ medium
- .26 ~ large

etat carré : 0.1127

Selon le critère de cohen proche mais pas certain (peut être corrélation)