Lapage

Analyse des ventes du site marchand

Lapage

Analyse exploratoire

Transactions

- Corriger le format date
- Vérifier la présence de sessions de test

```
# Afficher un résumé du DF
data_transactions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 679532 entries, 0 to 679531
Data columns (total 4 columns):
 #   Column      Non-Null Count   Dtype 
---  ------      --------------   ----- 
 0   id_prod     679532 non-null  object
 1   date        679532 non-null  object
 2   session_id  679532 non-null  object
 3   client_id   679532 non-null  object
dtypes: object(4)
memory usage: 20.7+ MB
```

```
# Calculer les statistiques descriptives pour toutes les colonnes
data_transactions.describe(include='all')
```

	id_prod	date	session_id	client_id
count	679532	679532	679532	679532
unique	3267	679371	342316	8602
top	1_369	test_2021-03-01 02:30:02.237413	s_0	c_1609
freq	2252	13	200	25488

Tout semble indiquer que des tests (200 itérations) ont été réalisés sur la base de données.

- la session 's_0' est une session de test ;
- le produit 'T_0' est un produit de test ;
- les clients 'ct_0' et 'ct_1' sont des clients de test.

```
# Filtrer les transactions pour la session "s_0"
tests = data_transactions[data_transactions['session_id'] == 's_0']

# Calculer les statistiques descriptives pour toutes les colonnes
tests.describe(include='all')
```

	id_prod	date	session_id	client_id
count	200	200	200	200
unique	1	39	1	2
top	T_0	test_2021-03-01 02:30:02.237413	s_0	ct_0
freq	200	13	200	106

```
# Afficher les identifiants des clients pour la session "s_0"
print(tests['client_id'].unique())
```

```
['ct_0' 'ct_1']
```

DF final

- 679 332 transactions
- 3 266 produits vendus
- 8 600 clients actifs

```
# Supprimer les lignes correspondant à des tests
data_transactions = data_transactions[data_transactions['session_id'] != 's_0']

# Convertir la colonne 'date' en format datetime
data_transactions['date'] = pd.to_datetime(data_transactions['date'], format='%Y-%m-%d')

# Ajouter des colonnes liées au temps pour l'agrégation
data_transactions['year_month'] = data_transactions['date'].dt.to_period('M')
data_transactions['day'] = data_transactions['date'].dt.to_period('D')

# Trier le DF par date en ordre chronologique
data_transactions.sort_values('date', inplace=True)
```

```
# Afficher des statistiques descriptives pour toutes les colonnes, y compris les colonnes datetime
data_transactions.describe(include='all', datetime_is_numeric=True)
```

	id_prod	date	session_id	client_id	year_month	day
count	679332	679332	679332	679332	679332	679332
unique	3266	NaN	342315	8600	24	730
top	1_369	NaN	s_118668	c_1609	2021-09	2022-11-30
freq	2252	NaN	14	25488	33326	1311
mean	NaN	2022-03-03 15:13:19.307389696	NaN	NaN	NaN	NaN
min	NaN	2021-03-01 00:01:07.843138	NaN	NaN	NaN	NaN
25%	NaN	2021-09-08 09:14:25.055994368	NaN	NaN	NaN	NaN
50%	NaN	2022-03-03 07:50:20.817730560	NaN	NaN	NaN	NaN
75%	NaN	2022-08-30 23:57:08.555173888	NaN	NaN	NaN	NaN
max	NaN	2023-02-28 23:58:30.792755	NaN	NaN	NaN	NaN

Products

- Corriger le format de categ
- 3 286 références

```
# Afficher un résumé du DF
data_products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3286 entries, 0 to 3286
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   id_prod  3286 non-null   object
 1   price    3286 non-null   float64
 2   categ    3286 non-null   int64
dtypes: float64(1), int64(1), object(1)
memory usage: 102.7+ KB
```

```
# Convertir la colonne 'categ' en type str
data_products['categ'] = data_products['categ'].astype(str)

# Afficher des statistiques descriptives pour toutes les colonnes
data_products.describe(include='all')
```

	id_prod	price	categ
count	3286	3286.000000	3286
unique	3286	NaN	3
top	0_1421	NaN	0
freq	1	NaN	2308
mean	NaN	21.863597	NaN
std	NaN	29.849786	NaN
min	NaN	0.620000	NaN
25%	NaN	6.990000	NaN
50%	NaN	13.075000	NaN
75%	NaN	22.990000	NaN
max	NaN	300.000000	NaN

➢ 3286 produit, contre 3267 dans les transactions, ce qui signifie qu'il y a des produits non vendus.

Customers

- 8 621 clients inscrits
- Ajouter l'âge pour les analyses

```
# Afficher un résumé du DF
data_customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8621 entries, 0 to 8622
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   client_id  8621 non-null   object
 1   sex        8621 non-null   object
 2   birth      8621 non-null   int64 
dtypes: int64(1), object(2)
memory usage: 269.4+ KB
```

```
# Afficher des statistiques descriptives pour toutes les colonnes
data_customers.describe(include='all')
```

	client_id	sex	birth
count	8621	8621	8621.000000
unique	8621	2	NaN
top	c_4410	f	NaN
freq	1	4490	NaN
mean	NaN	NaN	1978.275606
std	NaN	NaN	16.917958
min	NaN	NaN	1929.000000
25%	NaN	NaN	1966.000000
50%	NaN	NaN	1979.000000
75%	NaN	NaN	1992.000000
max	NaN	NaN	2004.000000

```
# Calculer l'âge des clients en se basant sur l'année de naissance
data_customers['age'] = pd.to_datetime('today').year - data_customers['birth']

# Ajouter une catégorie par tranche d'âge de 5 ans avec une précision de 0
data_customers['age_categ'] = pd.cut(data_customers['age'], 15, precision=0)
```

Regroupement des données

```
# Regrouper toutes les tables
data = data_transactions.merge(data_products, on='id_prod', how='left').merge(data_customers, on='client_id', how='left')
```

```
# Afficher des statistiques descriptives pour toutes les colonnes
data.describe(include='all', datetime_is_numeric=True)
```

	id_prod	date	session_id	client_id	year_month	day	price	categ	sex	birth	age	age_categ
count	679332	679332	679332	679332	679332	679332	679111.000000	679111	679332	679332.000000	679332.000000	679332
unique	3266	NaN	342315	8600	24	730	NaN	3	2	NaN	NaN	15
top	1_369	NaN	s_118668	c_1609	2021-09	2022-11-30	NaN	0	m	NaN	NaN	(39.0, 44.0]
freq	2252	NaN	14	25488	33326	1311	NaN	415459	340930	NaN	NaN	132679
mean	NaN	2022-03-03 15:13:19.307389696	NaN	NaN	NaN	NaN	17.454773	NaN	NaN	1977.811139	45.188861	NaN
min	NaN	2021-03-01 00:01:07.843138	NaN	NaN	NaN	NaN	0.620000	NaN	NaN	1929.000000	19.000000	NaN
25%	NaN	2021-09-08 09:14:25.055994368	NaN	NaN	NaN	NaN	8.870000	NaN	NaN	1970.000000	36.000000	NaN
50%	NaN	2022-03-03 07:50:20.817730560	NaN	NaN	NaN	NaN	13.990000	NaN	NaN	1980.000000	43.000000	NaN
75%	NaN	2022-08-30 23:57:08.555173888	NaN	NaN	NaN	NaN	18.990000	NaN	NaN	1987.000000	53.000000	NaN
max	NaN	2023-02-28 23:58:30.792755	NaN	NaN	NaN	NaN	300.000000	NaN	NaN	2004.000000	94.000000	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	18.328998	NaN	NaN	13.574553	13.574553	NaN

Valeur manquantes :

- Il y a 221 valeurs nulles pour les variables :
  - price
  - categ

➢ Un seul produit concerné ‘0_2245’
➢ Peu de ventes par rapport au volume de données

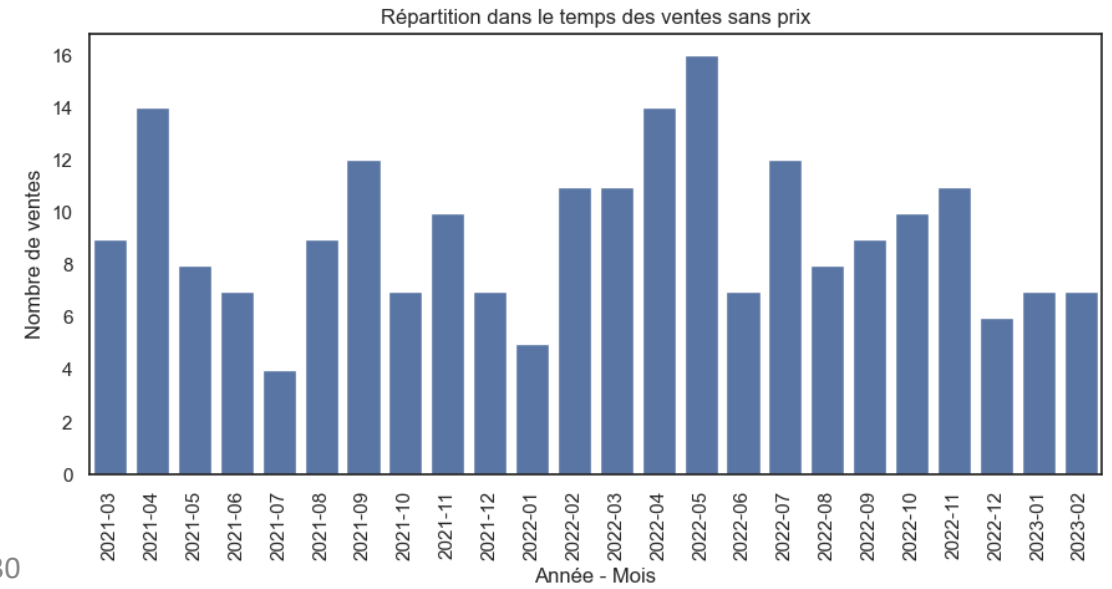
```
# Chercher les codes produits concernés
data.loc[data['price'].isnull(), 'id_prod'].unique()
```

```
array(['0_2245'], dtype=object)
```

```
# Vérifier la présence du produit dans la table products
data_products[data_products['id_prod'] == '0_2245']
```

id_prod	price	categ

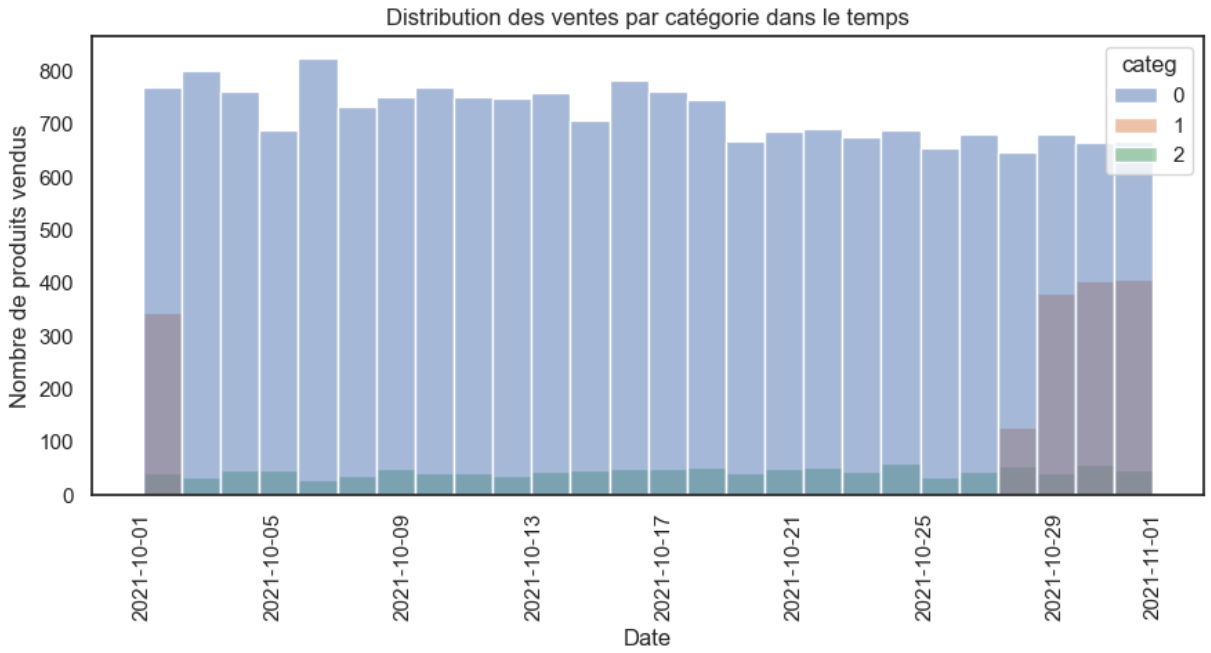
```
# Impact des ventes sans prix
plt.figure(figsize=(9, 5))
sns.countplot(data=data[data['price'].isnull()], x='year_month', color='b')
plt.title("Répartition dans le temps des ventes sans prix")
plt.xlabel("Année - Mois")
plt.ylabel("Nombre de ventes")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



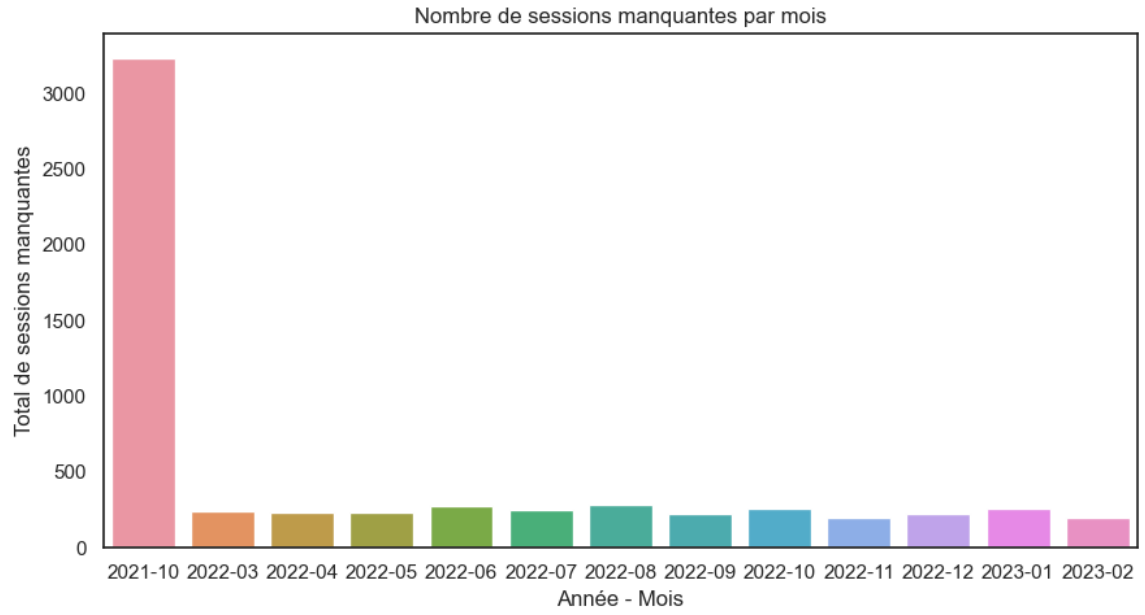
- Absence des ventes de la categorie 1 sur octobre 2021:

➢ On exclu ce mois dans l'analyse

```
# Visualiser la distribution des ventes par catégorie sur le mois d'octobre 2021
plt.figure()
plt.title("Distribution des ventes par catégorie dans le temps")
sns.histplot(data=data[data['year_month'] == '2021-10'], x='date', hue='categ')
plt.xlabel("Date")
plt.xticks(rotation=90)
plt.ylabel("Nombre de produits vendus")
plt.show()
```



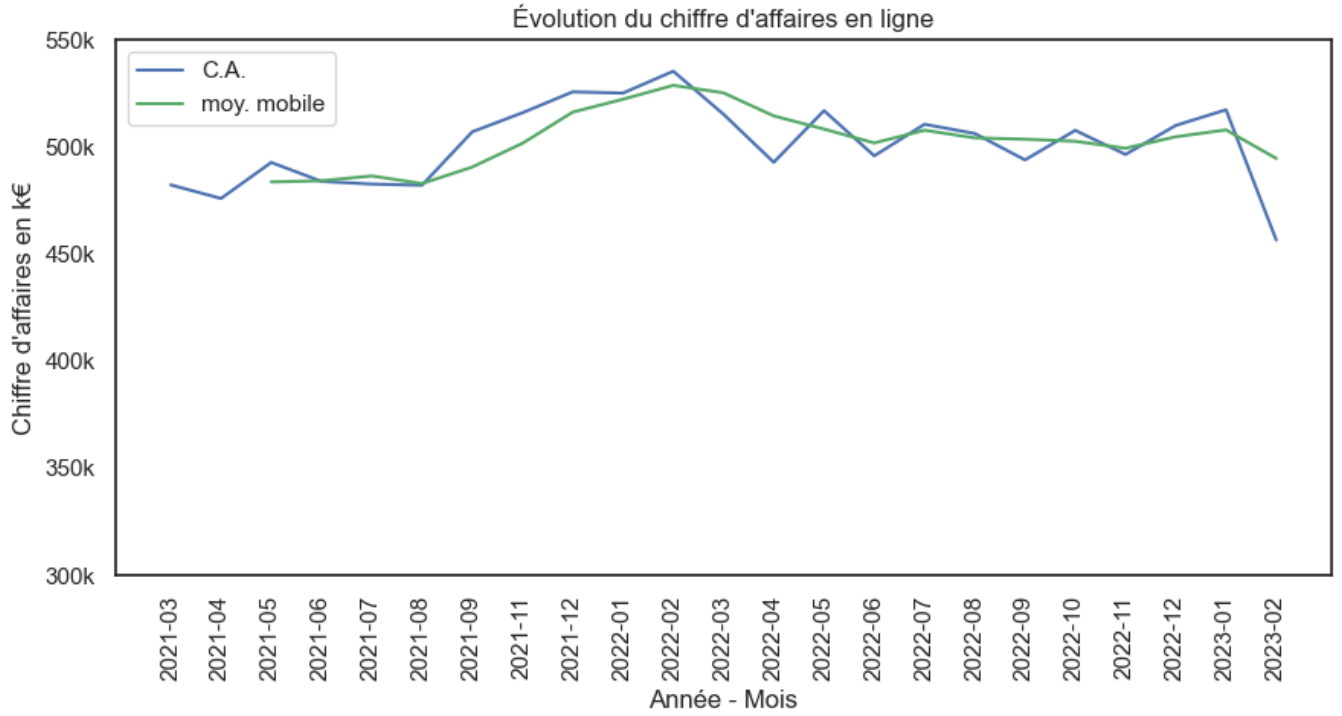

```
# Retirer 1 pour ne garder que les sessions manquantes.
# Ne pas tenir compte des écarts à 0 car cela correspond à la présence de plusieurs ventes sur une même session.
data.loc[data['session_var'] != 0, 'session_var'] = data.loc[data['session_var'] != 0, 'session_var'] - 1

# Visualiser le nombre de sessions manquantes cumulées par mois.
plt.figure()
plt.title("Nombre de sessions manquantes par mois")
sns.barplot(data=data[data['session_var'] > 0], x='year_month', y='session_var', estimator=sum, ci=False)
plt.xlabel("Année - Mois")
plt.ylabel("Total de sessions manquantes")
plt.show()
```



Analyse de chiffre d’affaires

Chiffre d’affaires

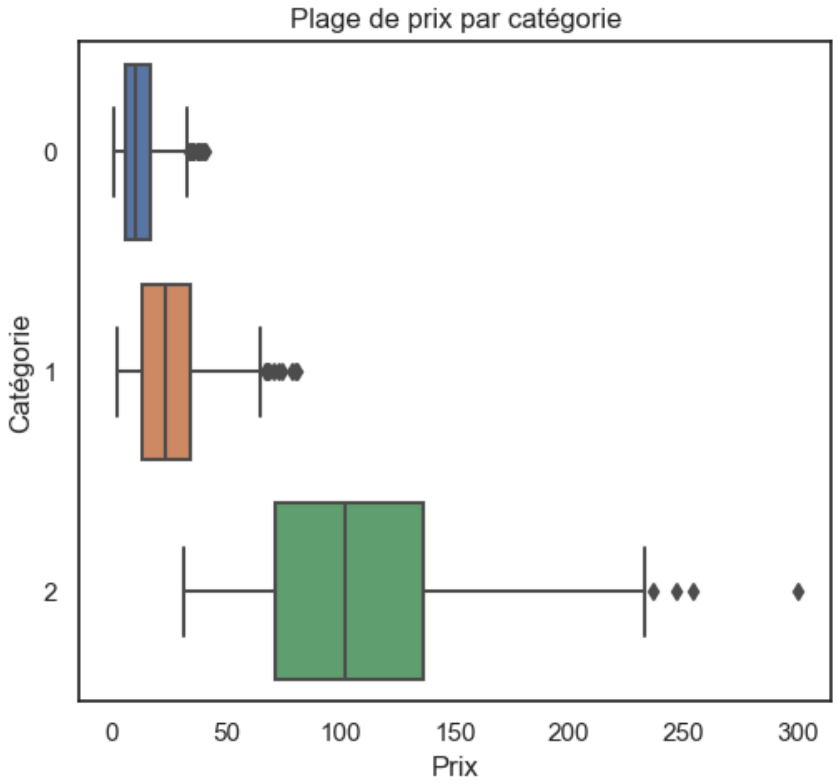
- 11,5 millions € sur 24 mois
- 6 millions € sur les 12 derniers mois

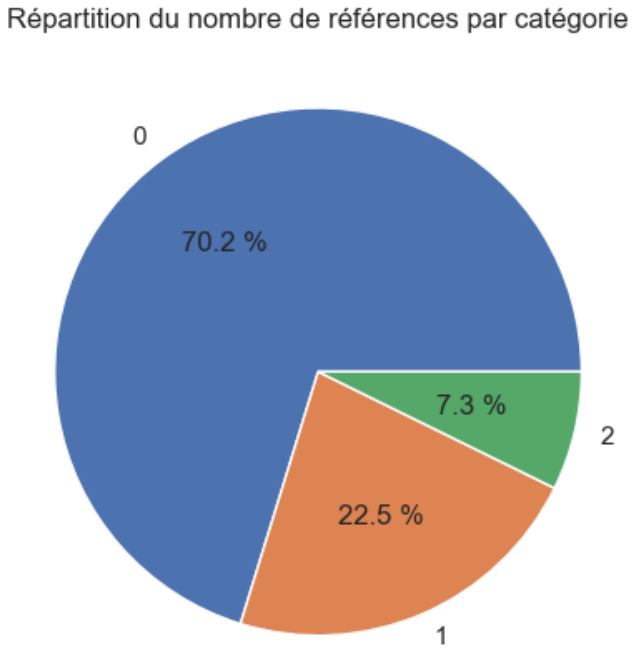


Analyse des ventes par produit

Un large choix

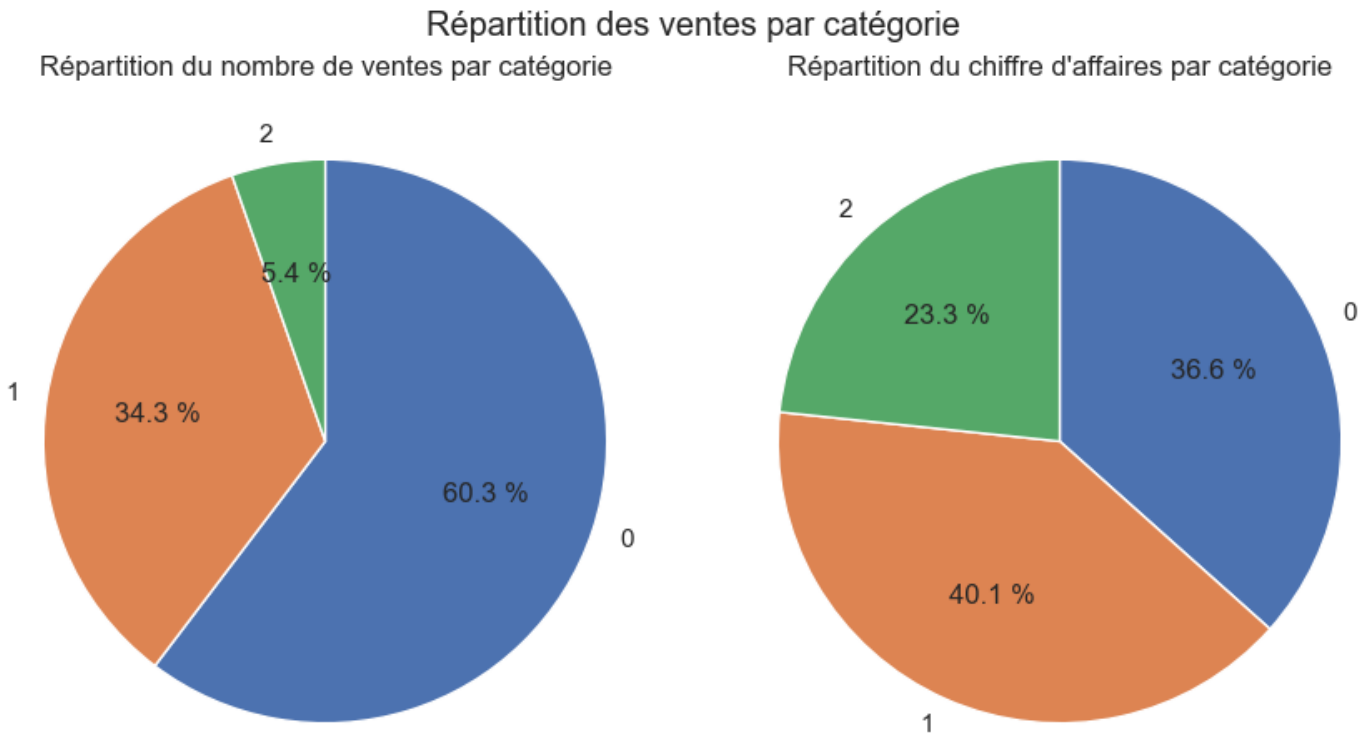
- 3 286 références





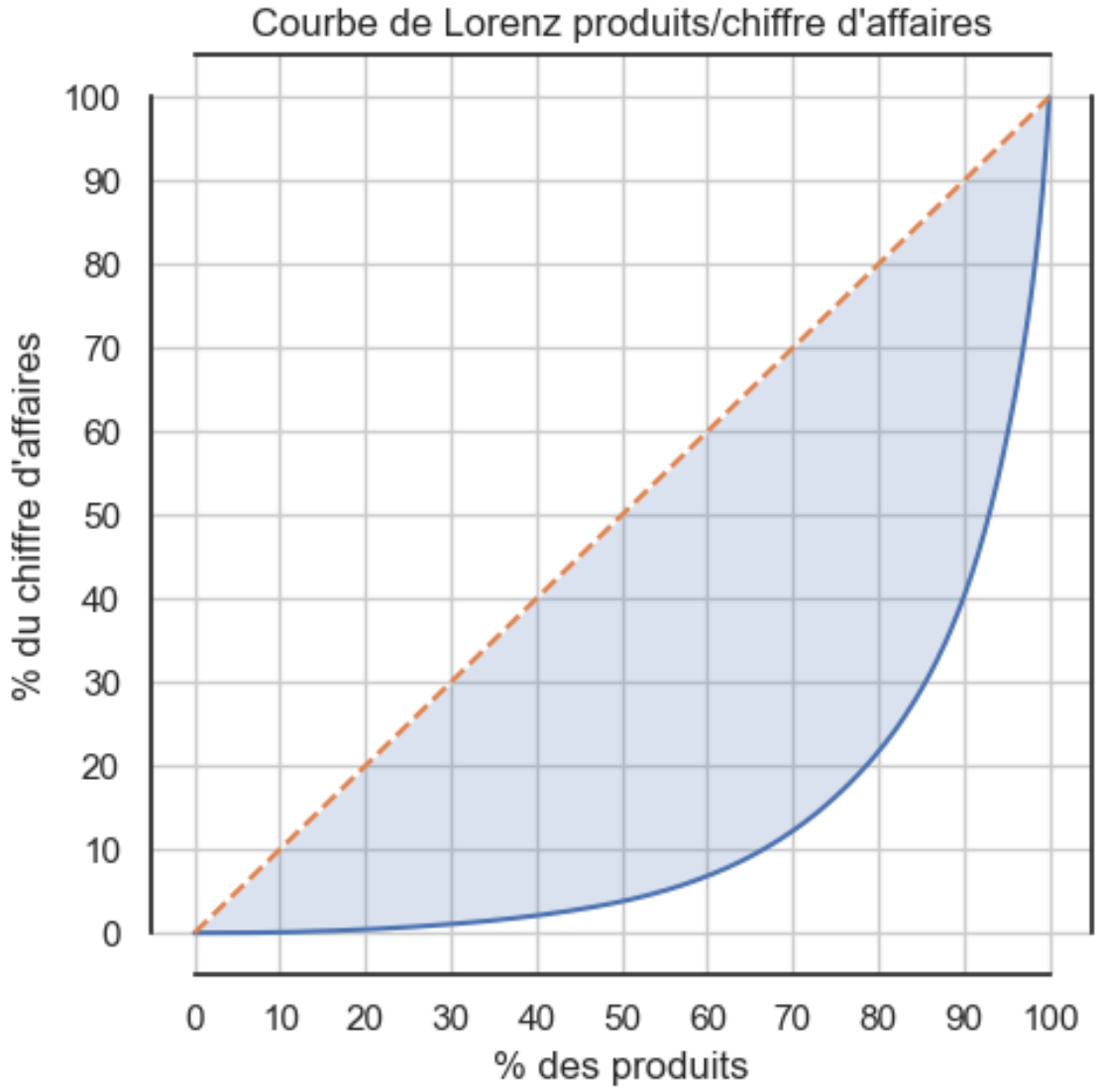
Par catégorie

- la catégorie 1 sur-performe avec 34% des ventes réalisées alors qu'elle ne représente que 22% des références
- C'est la catégorie 0 qui en pâtit avec 'seulement' 60% des ventes réalisées pour 70% des références présentes sur le site.



Par produit

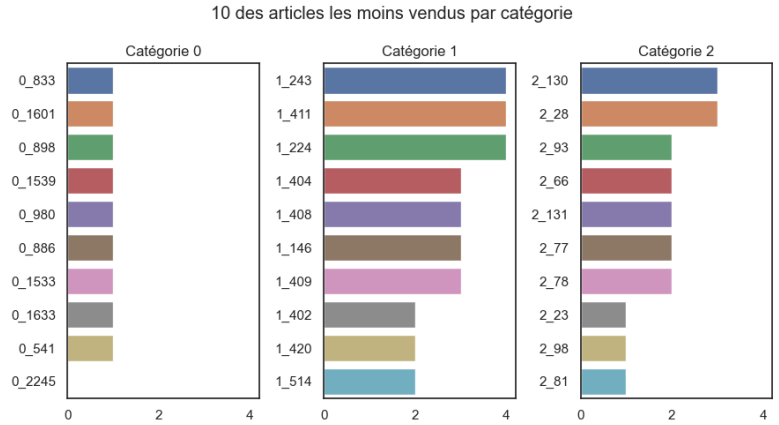
- 20 % des produits réalisent 80% du chiffre d'affaires
- Indice de Gini : 0,74

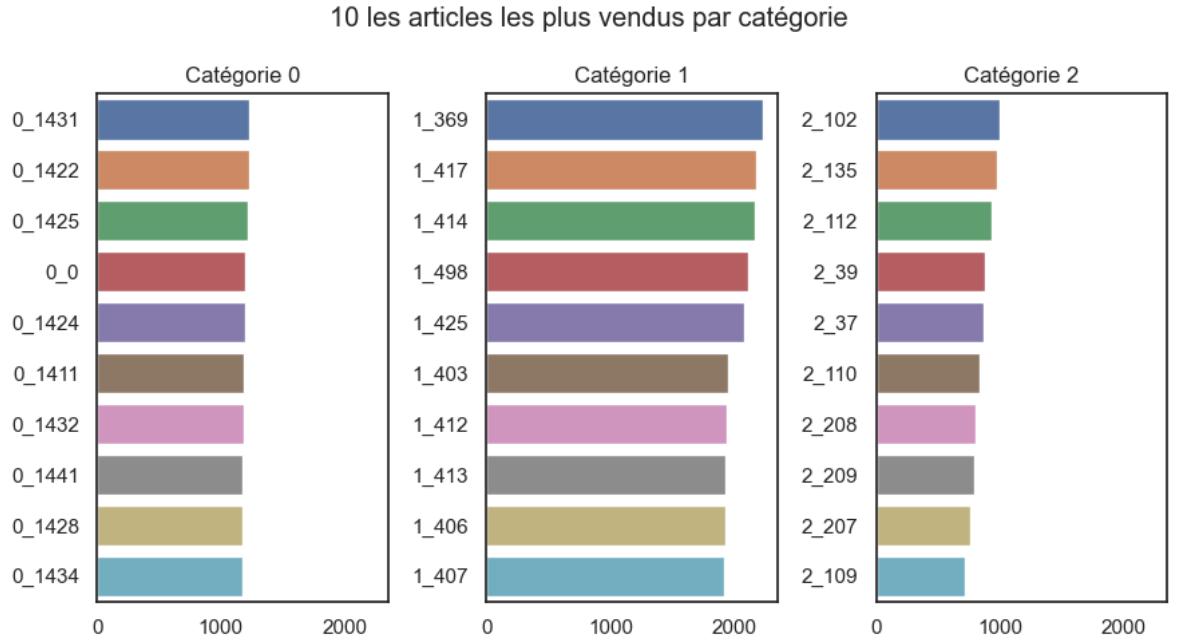


TOP 10 / FLOP 10

	index	id_prod	price	categ
0	811	0_1014	1.15	0
1	184	0_1016	35.06	0
2	1543	0_1025	24.99	0
3	737	0_1062	20.08	0
4	794	0_1119	2.99	0
5	1031	0_1318	20.92	0
6	1530	0_1620	0.80	0
7	2408	0_1624	24.50	0
8	1347	0_1645	2.99	0
9	279	0_1780	1.67	0
10	1139	0_1800	22.05	0
11	3096	0_2308	20.28	0
12	2690	0_299	22.99	0
13	2215	0_310	1.94	0
14	1505	0_322	2.99	0
15	3031	0_510	23.66	0
16	846	1_0	31.82	1
17	1863	1_394	39.73	1
18	1946	2_72	141.32	2
19	2525	2_86	132.36	2
20	1709	2_87	220.99	2

Il y a 21 produits invendus sur un total de 3286, soit 0.64%.

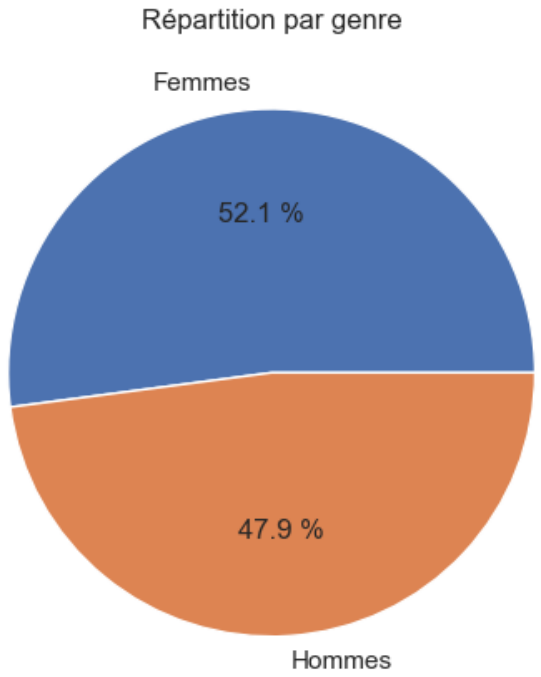


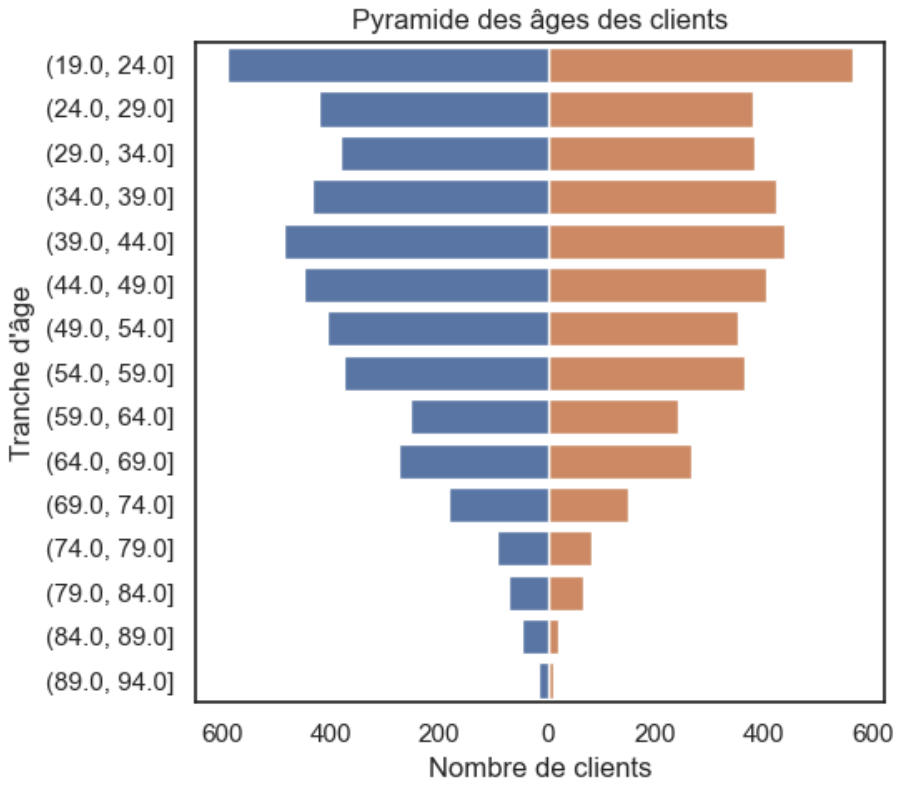


Analyse des ventes par client

Une clientèle jeune

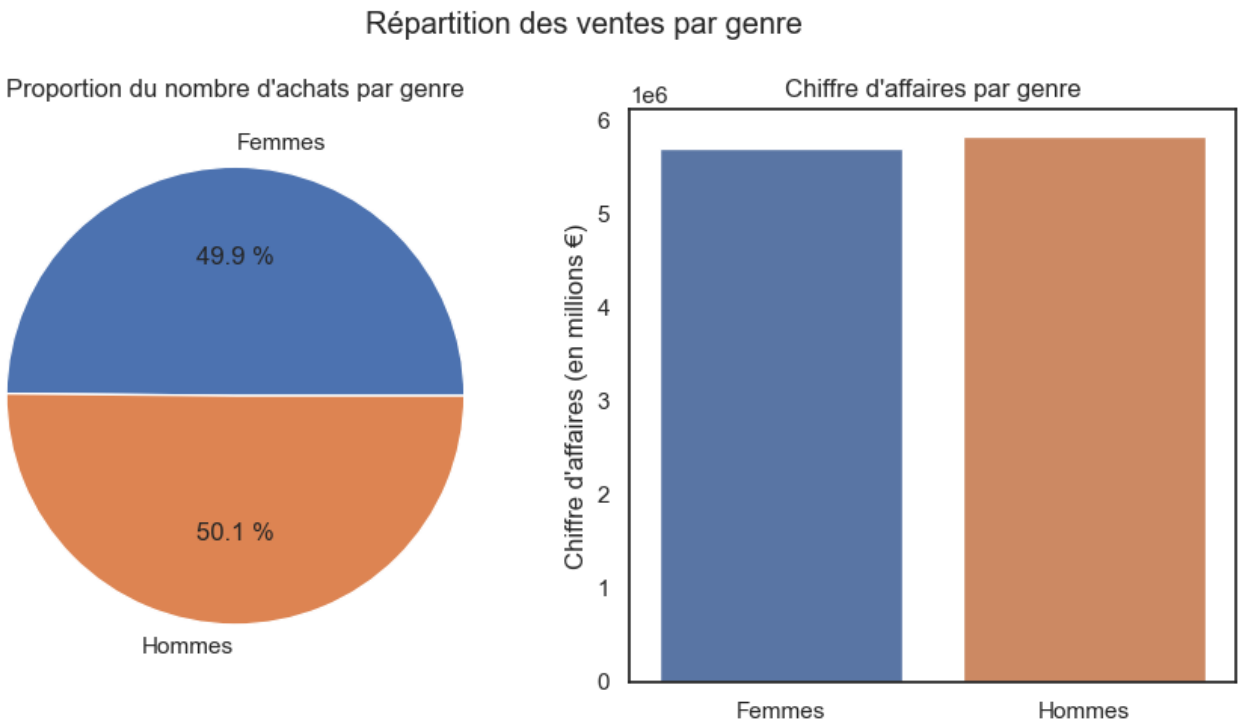
- 1 clients sur 2 a moins de 45 ans





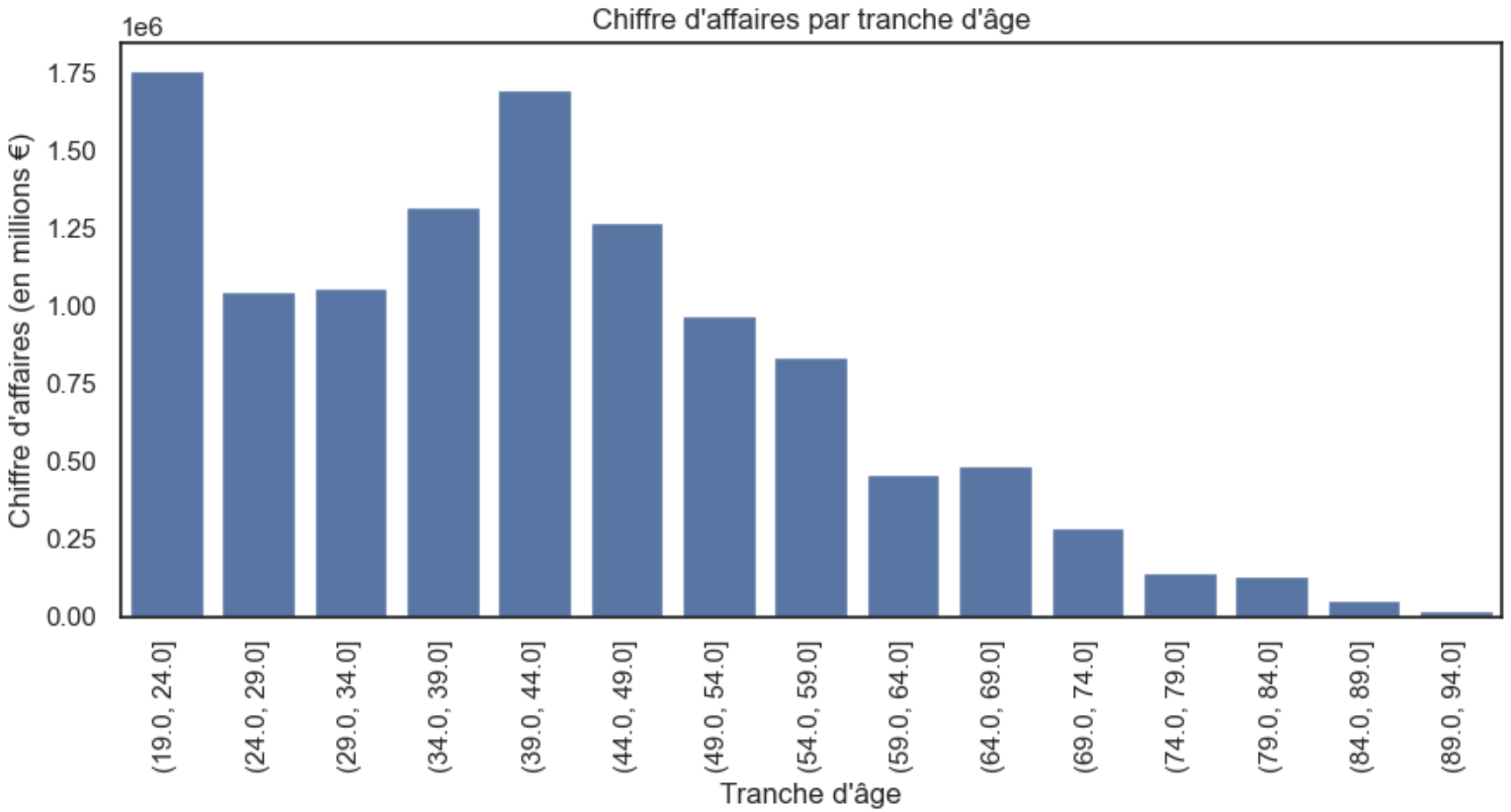
Par genre

Alors que les femmes sont légèrement plus nombreuse elle représentent tout juste 50 % des ventes.



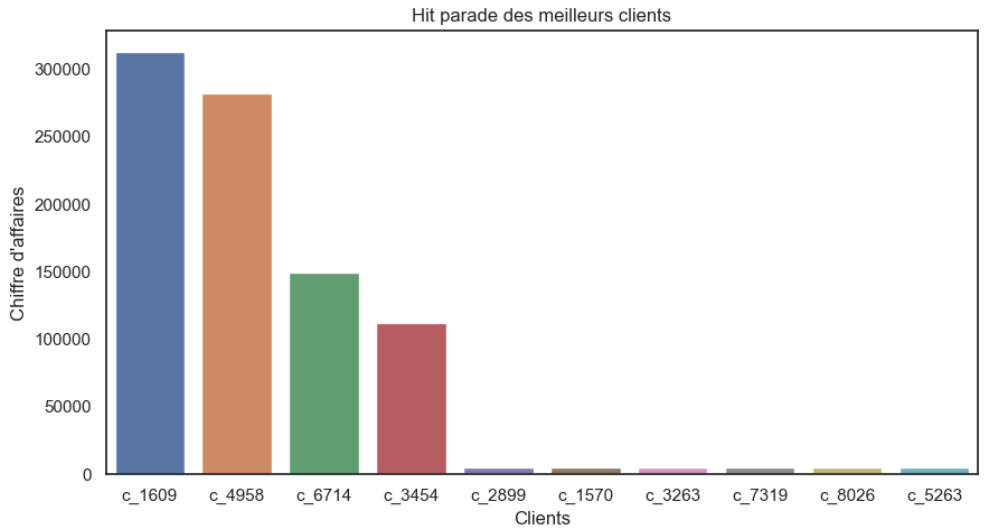
Par âge

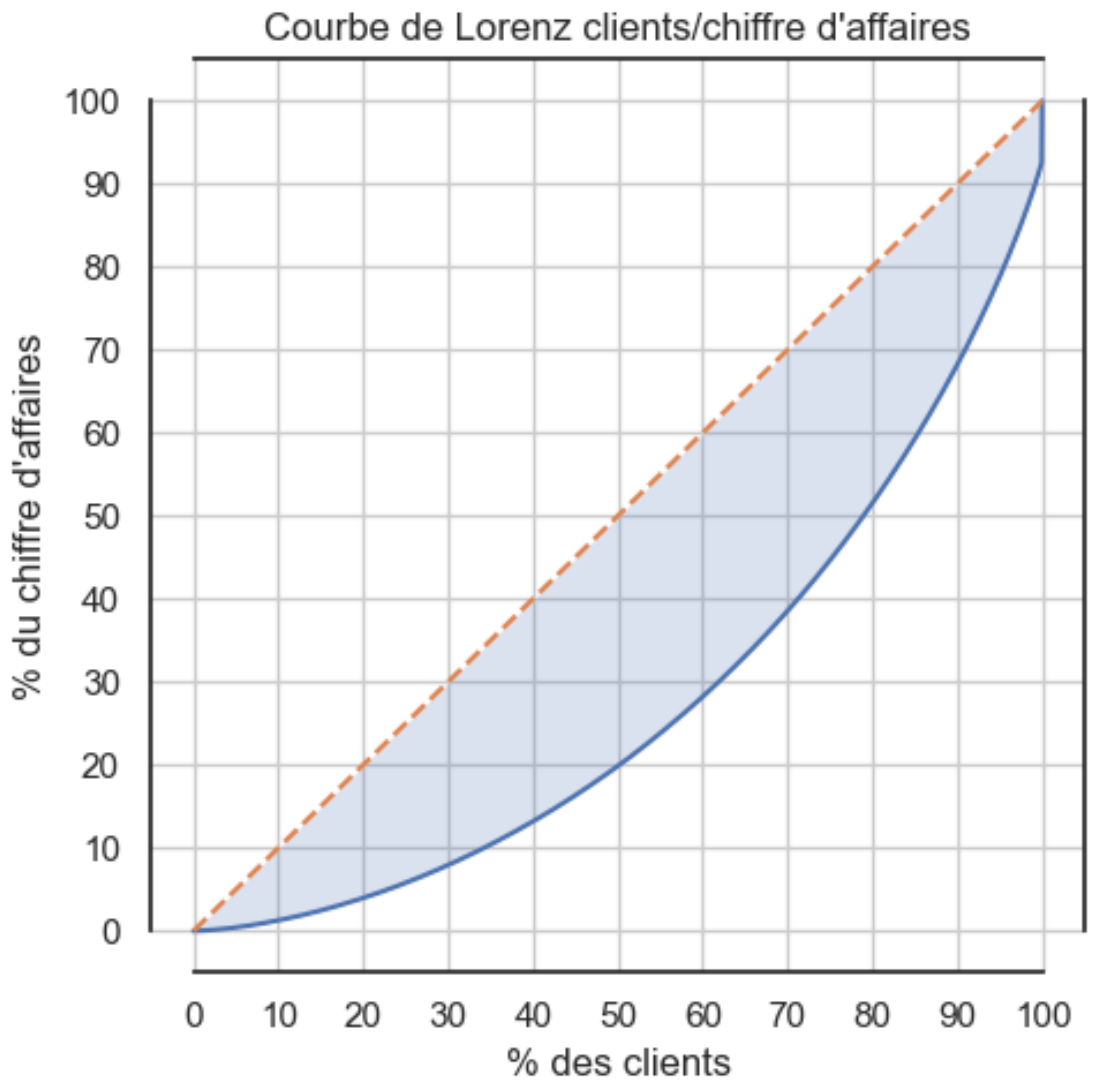
- 85 % du C.A est réalisé auprès des clients de moins de 60 ans
- Les moins de 30 ans représentent à eux seuls 25 % du C.A.



Répartiotion du chiffre d'affaires

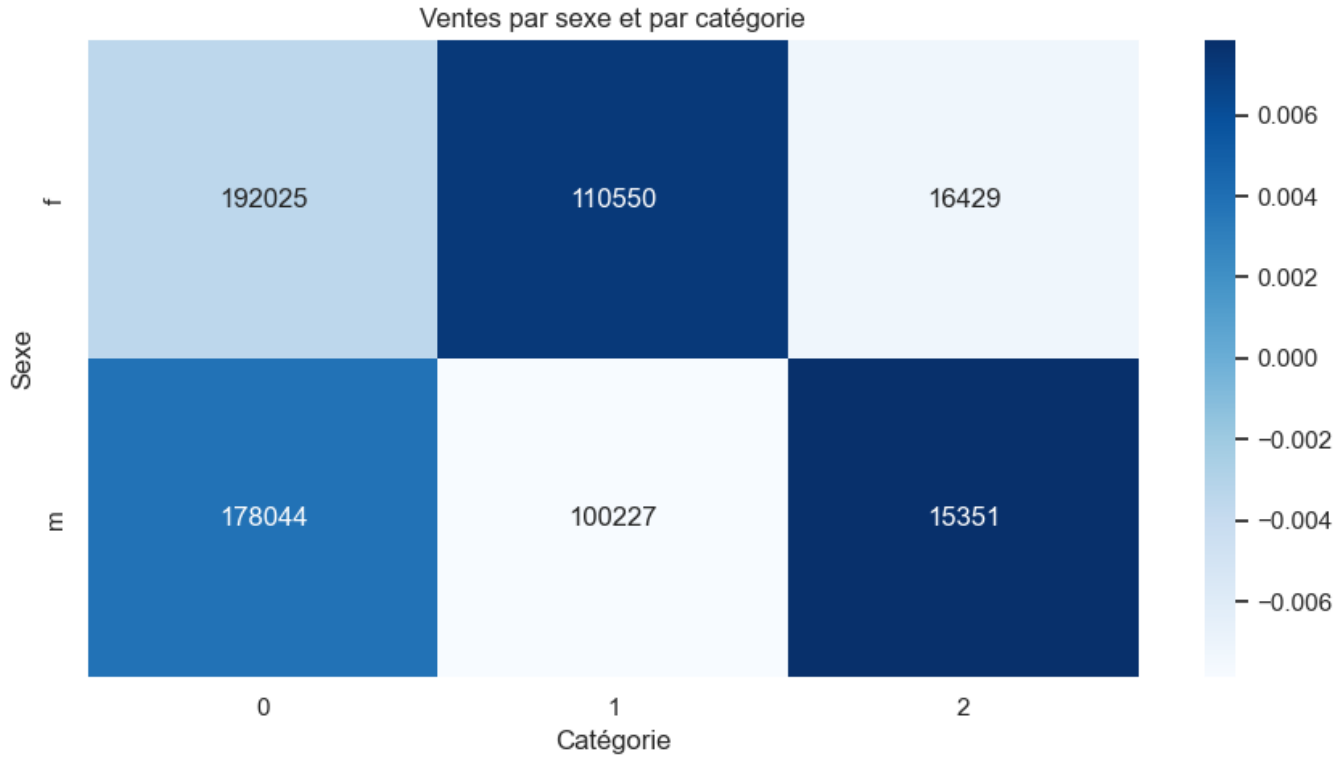
- Indice de Gini : 0,45
- 20 % des clients réalisent 50 % du C.A.
- Les 4 plus gros clients réalisent presque 10 % du C.A.:





Recherche de corrélations

Lien entre le genre et la catégorie



- 2 variables qualitatives
- Test du khi 2
- P-value = 0,0046

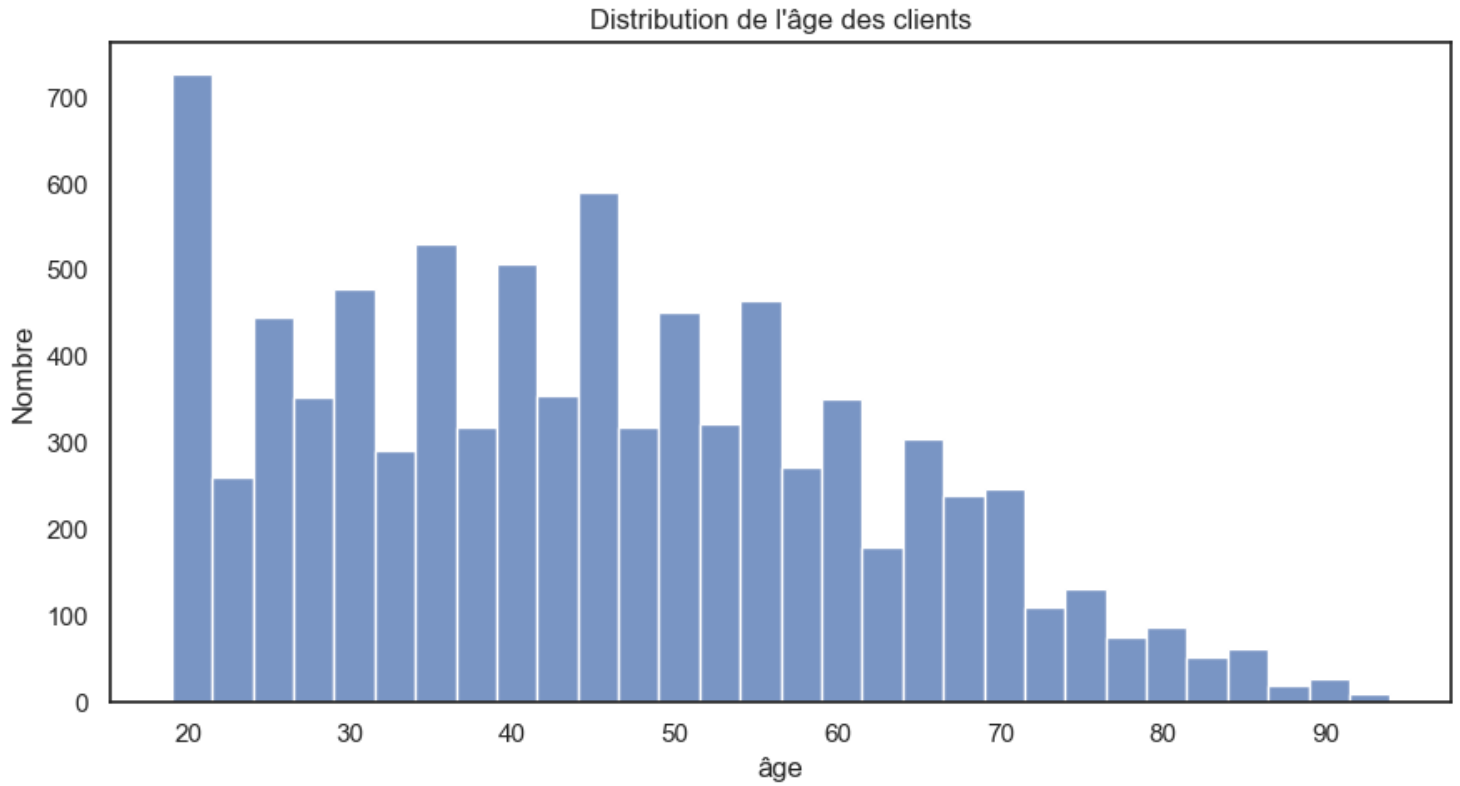
➢ Les deux variables ne sont pas indépendantes

Normalité de la variable âge

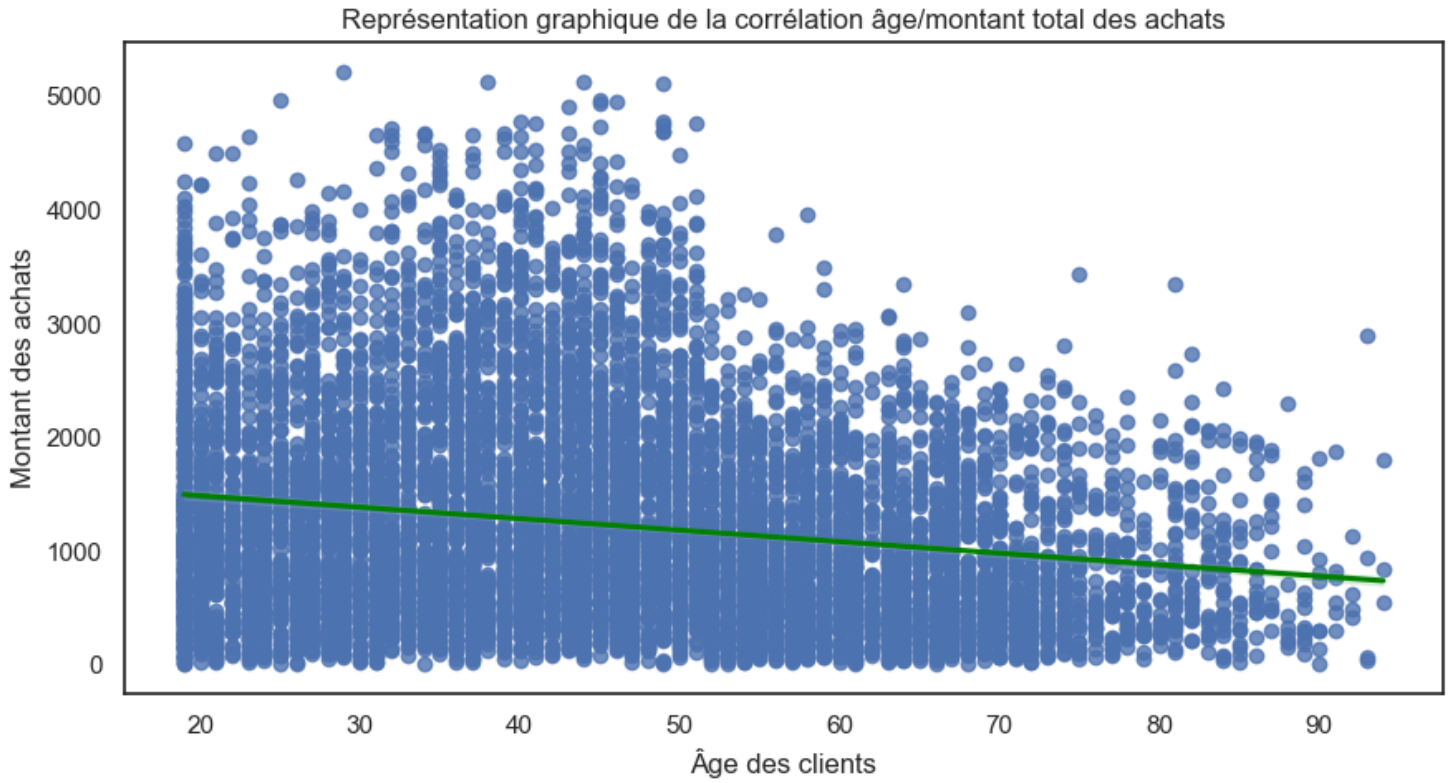
Âge

- Test de Kolmogorov-Smirnov
- P-value : 0

➢ La distribution n’est pas normale



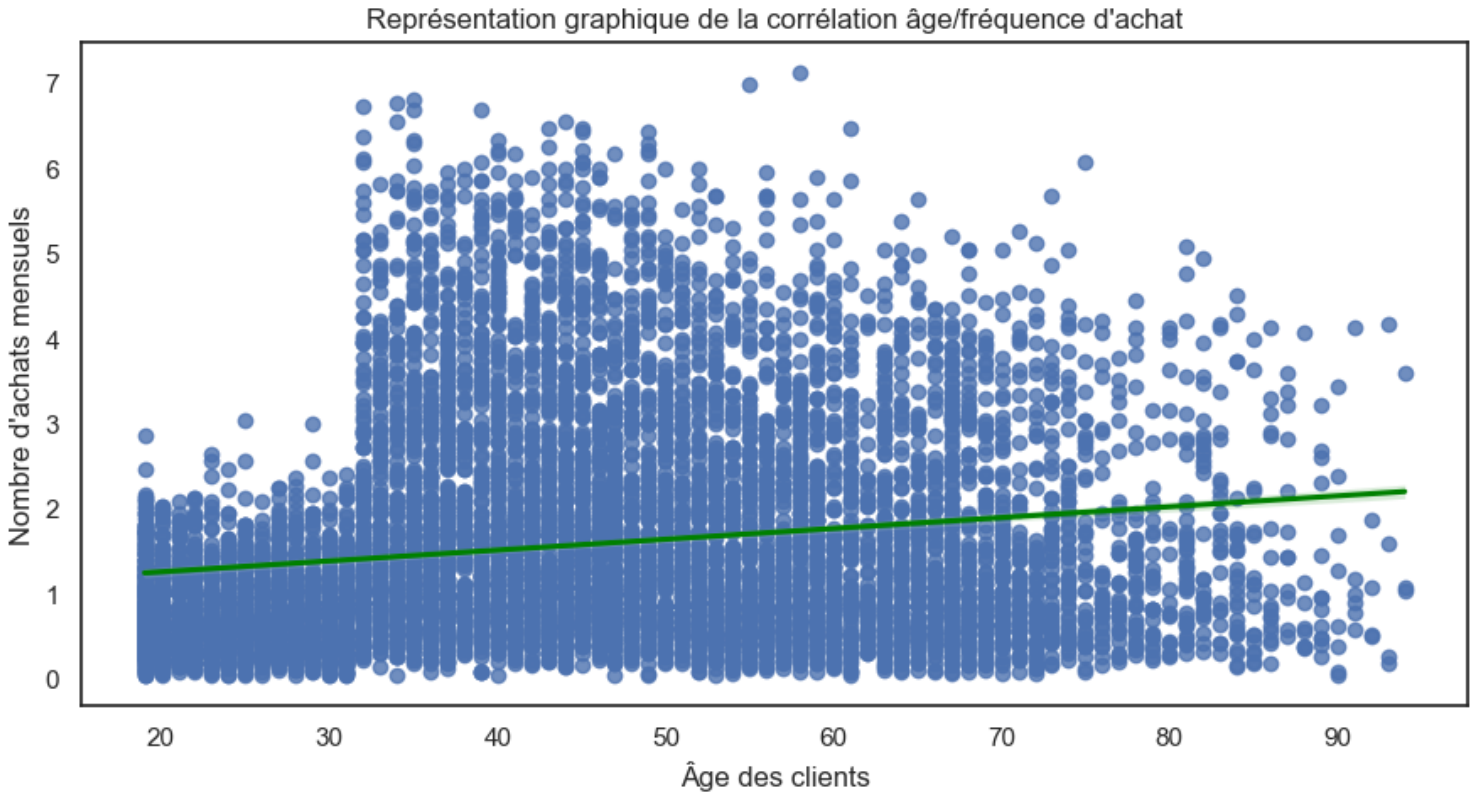
Lien entre l'âge et le montant total



- 2 variables quantitatives
- Test de Spearman (non paramétrique)
- P-value = 0,0
- Coefficient de correlation = -0,18

➢ Les deux variables ne sont pas indépendantes

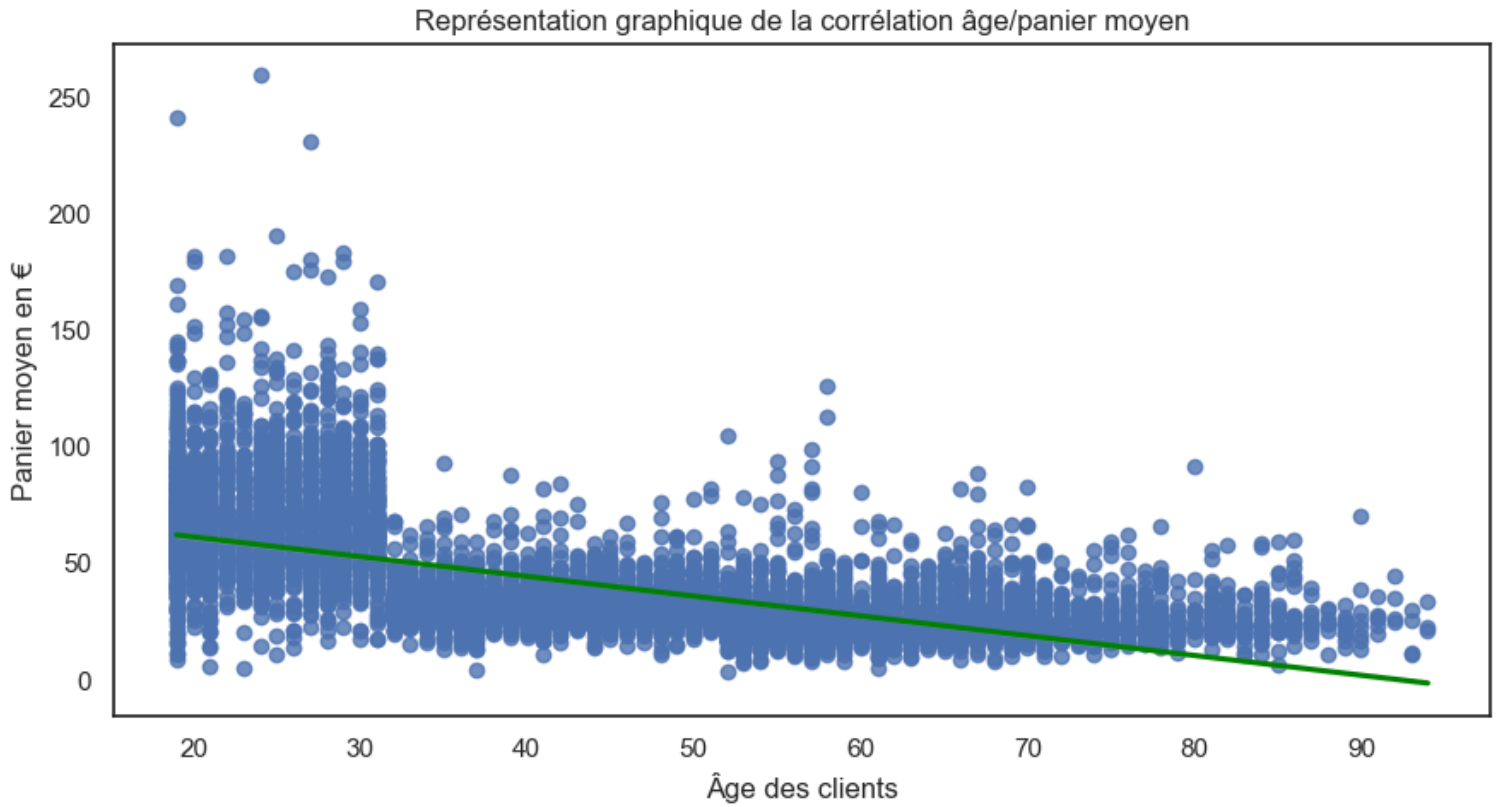
Lien entre l’âge et la fréquence d’achat



- 2 variables quantitatives
- Test de Spearman (non paramétrique)
- P-value = 0,0
- Coefficient de correlation = 0,21

➢ Les deux variables ne sont pas indépendantes

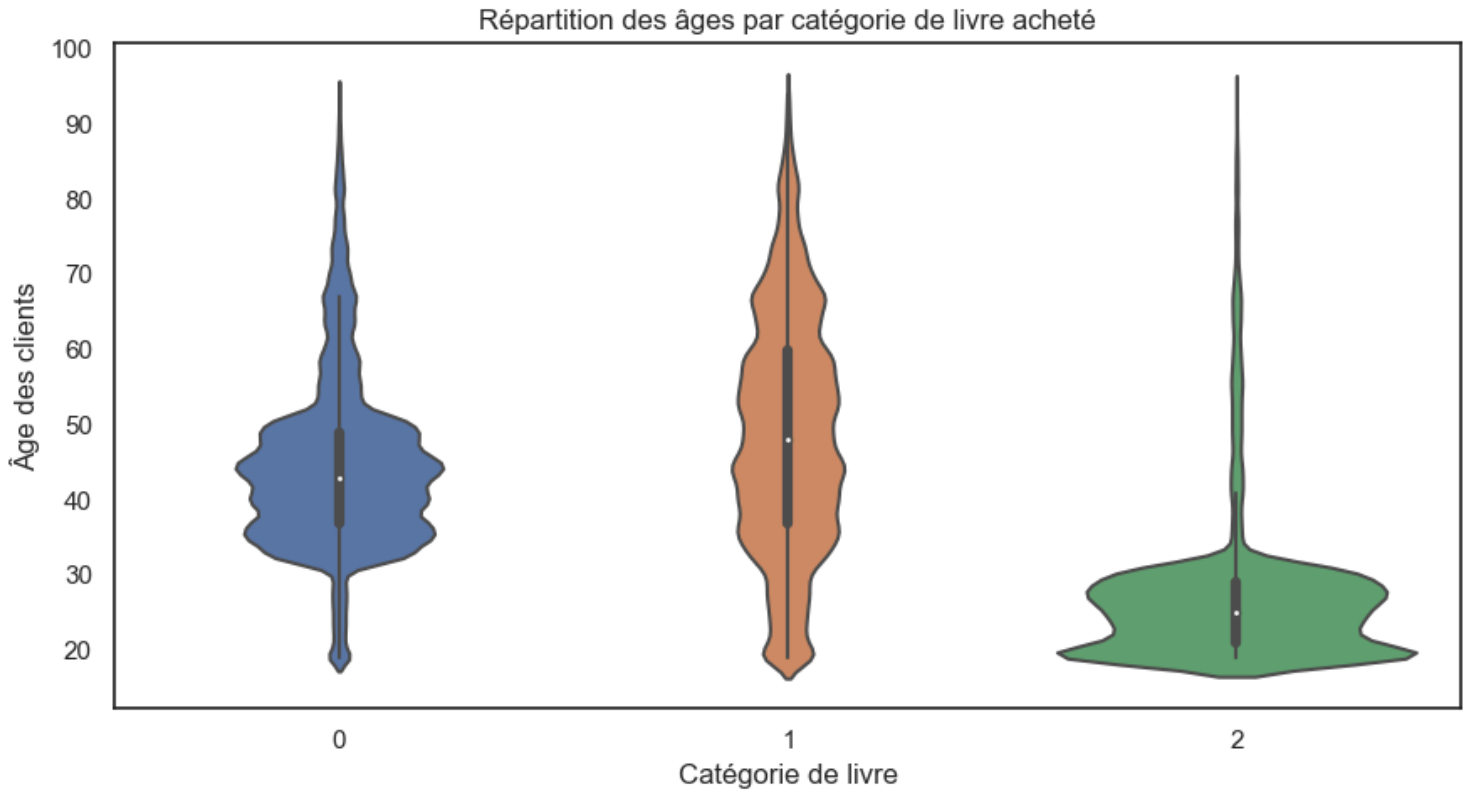
Lien entre l'âge et le panier moyen



- 2 variables quantitatives
- Test de Spearman (non paramétrique)
- P-value = 0,0
- Coefficient de correlation = -0,69

➢ Les deux variables ne sont pas indépendantes

Lien entre l'âge et la catégorie



- 1 variable qualitative, 1 variable quantitative
- Test de Kruskal-Wallis
- P-value = 0,0

➢ Les deux variables ne sont pas indépendantes