Lapage

Soutenance

P6: Analysez les ventes d'une librairie avec Python

Xiuting LIANG 05.04.2022

OpenClassrooms - "Île-de-France E-learning" 2021-2022 - Data Analyst

Sommaires

01

Détail du nettoyage des données

02

L'analyse de vente

- Le chiffre d'affaires (CA) et la tendance
- Un zoom sur les références
- Les profils des clients

03

Les corrélations

1. Nettoyage des données

products.csv

- pas de valeur manquante
- pas de valeur doublon
- les prix moyennes de chaque catégorie
- outliers des prix

- **traiter le prix inférieur à 0; (T_0)**

transactions.csv

- pas de valeur manquante

- **traiter les transactions pour le test (T_0)**

- **'date' to_datetime**

- l'attention à l'unicité

customers.csv

- pas de valeur manquante
- pas de valeur doublon
- ajouter la colonne 'age'
- pas de valeur aberrant

la jointure

- outer joint: 'df_trans' et 'df_product'
- left join: 'df_temp1' et 'df_cus'

- **traiter les valeurs manquantes (id_prod: 0_2245)**
- **calculer le chiffre d'affaire (CA) de chaque transaction**

Traiter le prix inférieur à 0 (T_0); 'date' to_datetime

```
df_product.describe()
```

	price	
count	3287.000000	3287.
mean	21.856641	0.
std	29.847908	0.
min	-1.000000	
25%	6.990000	0.
50%	13.060000	0.
75%	22.990000	1.
max	300.000000	2.

1.1.1. Traiter le prix inférieur à 0; (T_0)

En fonction des analyses suivants (dans la partie 1.2.1. traiter les d
notre analyse, il faut l'exclure.

```
# verifier le prix qui est inférieur à 0 euro
df_product.loc[df_product['price'] < 0, :]
```

	id_prod	price	categ
731	T_0	-1.0	0

```
# Exclure le produit avec le prix < 0 euro
df_product = df_product.loc[df_product['price'] >= 0, :]
df_product['price'].describe()
```

```
count    3286.000000
mean       21.863597
std        29.849786
min         0.620000
25%         6.990000
50%        13.075000
75%        22.990000
max       300.000000
Name: price, dtype: float64
```

```
df_trans.describe()
```

	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

```
df_null_time['id_prod'].describe()
```

```
count     200
unique      1
top       T_0
freq      200
Name: id_prod, dtype: object
```

	id_prod	date	session_id	client_id	date_test
3019	T_0	test_2021-03-01 02:30:02.237419	s_0	ct_0	NaT
5138	T_0	test_2021-03-01 02:30:02.237425	s_0	ct_0	NaT
9668	T_0	test_2021-03-01 02:30:02.237437	s_0	ct_1	NaT
10728	T_0	test_2021-03-01 02:30:02.237436	s_0	ct_0	NaT
15292	T_0	test_2021-03-01 02:30:02.237430	s_0	ct_0	NaT

*Comme discuté dans la partie "1.1.1. La dispersion de données? Valeur aberrant?", les 200 lignes de transactions avec le même id_prod, et les temps de transactions presque pareils, sont les transactions génerés pour le test de système par les développeurs. Pour notre analyse, il faut les supprimer.

```
# supprimer les valeurs manquants pour la colonne ['date_test'] pour supprimer les valeurs anormaux
df_trans = df_trans.dropna(subset = ['date_test'])
any(df_trans['date_test'].isnull()) # verifier s'il y a autre ligne avec valeurs anormaux
```

```
False
```

```
# transferer la format de 'date' à datetime
df_trans['date'] = pd.to_datetime(df_trans['date'])
df_trans.info()
```

Traiter les valeurs manquantes (id_prod: 0_2245)

```
# selectionner les lignes avec prix manquantes
df_null = df.loc[df['price'].isnull(),:]

# verifier l'information de cettes lignes
df_null[['id_prod','session_id','client_id','sex']].describe()
```

	id_prod	session_id	client_id	sex
count	221	221	221	221
unique	1	221	100	2
top	0_2245	s_272266	c_1533	f
freq	221	1	6	117

```
# verifier s'il y a 'id_prod: 0_2245' dans la liste de produit
any(df_product['id_prod']=='0_2245')
```

```
False
```

```
df_product.head(12)
```

	id_prod	price	categ
0	0_1421	19.99	0
1	0_1368	5.13	0
2	0_731	17.99	0
3	1_587	4.99	1

*le premier numéro (0,1,2) de 'id_prod' indique la catégorie de chaque produit (0,1,2). '0_2245' est dans la categ 0.

```
# ajouter le catégory '0' pour le produit '0_2245'
df['categ'] = df['categ'].fillna(0)
```

*imputer le prix de '0_2245' par 11.73 euro, le prix moyenne de categ 0

1.5.2. imputer les prix pour le produit '0_2245'

```
# calculé déjà les prix moyenne de chaque catégorie dans la
df_cmean
```

categ	price
0	11.727280
1	25.531421
2	108.354686

***imputer le prix de '0_2245' par 11.73 euro, le prix moyenne de categ 0**

```
[30]: # imputer les prix null avec le moyenne de prix de tous les livres
df['price'] = df['price'].fillna(11.73)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 679353 entries, 0 to 679352
Data columns (total 9 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   id_prod     679353 non-null  object
 1   date        679332 non-null  datetime64[ns]
 2   session_id  679332 non-null  object
 3   client_id   679332 non-null  object
 4   price       679353 non-null  float64
 5   categ       679353 non-null  float64
 6   sex         679332 non-null  object
 7   birth       679332 non-null  float64
 8   age         679332 non-null  float64
dtypes: datetime64[ns](1), float64(4), object(4)
memory usage: 51.8+ MB
```

Les autres traitements

1.4. La jointure de 'df_product', 'df_trans', 'df_cus'

***outer join 'df_trans' et 'df_product': pour garder tous les produits (même si les 0 vente) et tous les transactions.**

***left join 'df_temp1' et 'df_cus': pour garder tous les produits (même si les 0 vente) et tous les transactions.**

```
# la jointure de 3 dateframes avec outer joint et left joint
df_temp1 = pd.merge(df_trans, df_product, on = 'id_prod', how = 'outer')
df = pd.merge(df_temp1, df_cus, on = 'client_id', how = 'left')
df.index.name='index'
df.head()
```

index	id_prod	date	session_id	client_id	price	categ	sex	birth	age
0	0_1518	2022-05-20 13:21:29.043970	s_211425	c_103	4.18	0.0	f	1986.0	36.0
1	0_1518	2021-09-26 12:37:29.780414	s_95811	c_6197	4.18	0.0	m	1985.0	37.0
2	0_1518	2021-05-06 17:14:43.117440	s_30782	c_682	4.18	0.0	f	1974.0	48.0
3	0_1518	2022-03-16 18:57:10.420103	s_180057	c_5932	4.18	0.0	f	1962.0	60.0
4	0_1518	2022-11-12 18:58:10.574853	s_296584	c_7217	4.18	0.0	f	1976.0	46.0

1.6. calculer le chiffre d'affaire (CA) de chaque transaction

```
df['CA'] = df.loc[df['session_id'].notnull(), 'price']
df['CA'] = df['CA'].fillna(0)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 679353 entries, 0 to 679352
Data columns (total 10 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   id_prod     679353 non-null  object
 1   date        679332 non-null  datetime64[ns]
 2   session_id  679332 non-null  object
 3   client_id   679332 non-null  object
 4   price       679353 non-null  float64
 5   categ       679353 non-null  float64
 6   sex         679332 non-null  object
 7   birth       679332 non-null  float64
 8   age         679332 non-null  float64
 9   CA          679353 non-null  float64
dtypes: datetime64[ns](1), float64(5), object(4)
memory usage: 57.0+ MB
```

2.0.2. Traiter les formats de datetime

```
df['year'] = df['date'].dt.strftime("%Y")
df['month'] = df['date'].dt.strftime("%Y-%m")
df['short_date'] = df['date'].dt.strftime("%Y-%m-%d")
df['quarter'] = df['date'].dt.to_period('Q')
df['day'] = df['date'].dt.day_name()
df.head()
```

Traiter dans la 2e partie après l'importation de donnés

index	id_prod	date	session_id	client_id	price	categ	sex	birth	age	CA	year	month	short_date	quarter	day
0	0_1518	2022-05-20 13:21:29.043970	s_211425	c_103	4.18	0	f	1986.0	36.0	4.18	2022	2022-05	2022-05-20	2022Q2	Friday
1	0_1518	2021-09-26 12:37:29.780414	s_95811	c_6197	4.18	0	m	1985.0	37.0	4.18	2021	2021-09	2021-09-26	2021Q3	Sunday
2	0_1518	2021-05-06 17:14:43.117440	s_30782	c_682	4.18	0	f	1974.0	48.0	4.18	2021	2021-05	2021-05-06	2021Q2	Thursday
3	0_1518	2022-03-16 18:57:10.420103	s_180057	c_5932	4.18	0	f	1962.0	60.0	4.18	2022	2022-03	2022-03-16	2022Q1	Wednesday
4	0_1518	2022-11-12 18:58:10.574853	s_296584	c_7217	4.18	0	f	1976.0	46.0	4.18	2022	2022-11	2022-11-12	2022Q4	Saturday

Les transactions manquants en 2021-10:

- Sera traiter prochainement d'après les besoins des analyses suivantes

Clients 18 ans

- discussion dans prochaine partie

L'analyse de vente

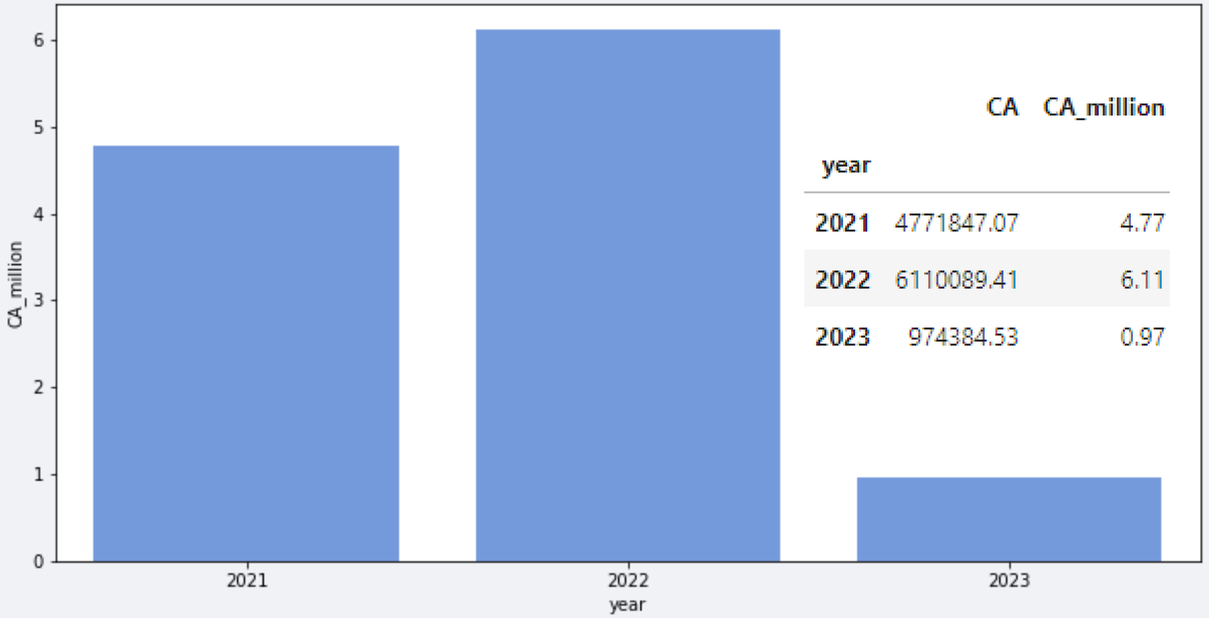
Le chiffre d'affaires (CA) et la tendance

Un zoom sur les références

Les profils des clients

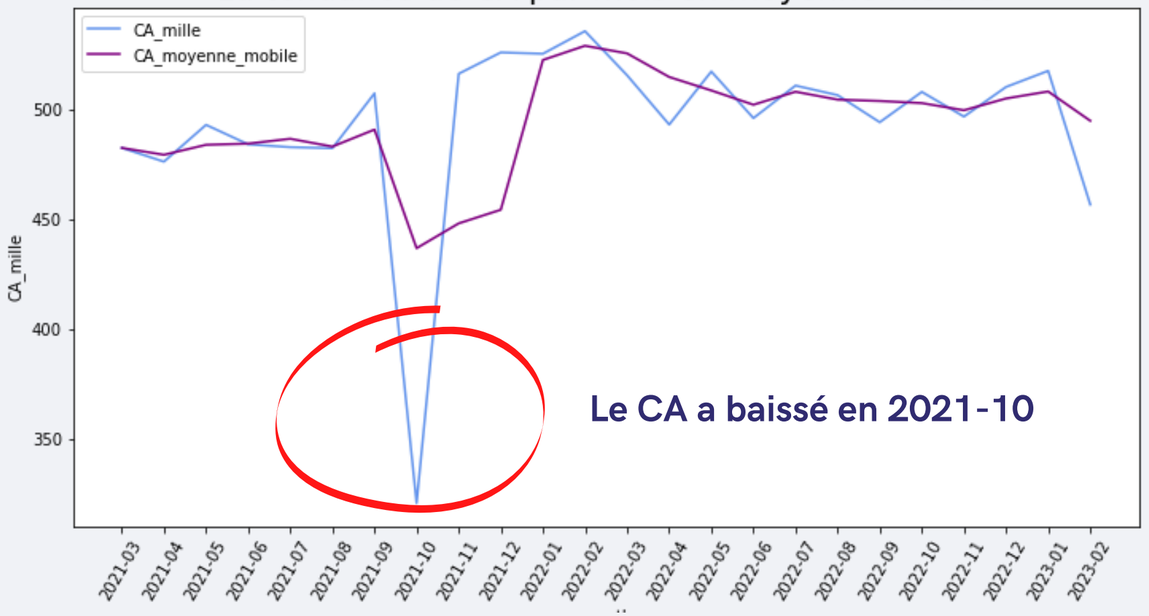
Le chiffre d'affaires (CA)

L'évolution du chiffre d'affaires annuel

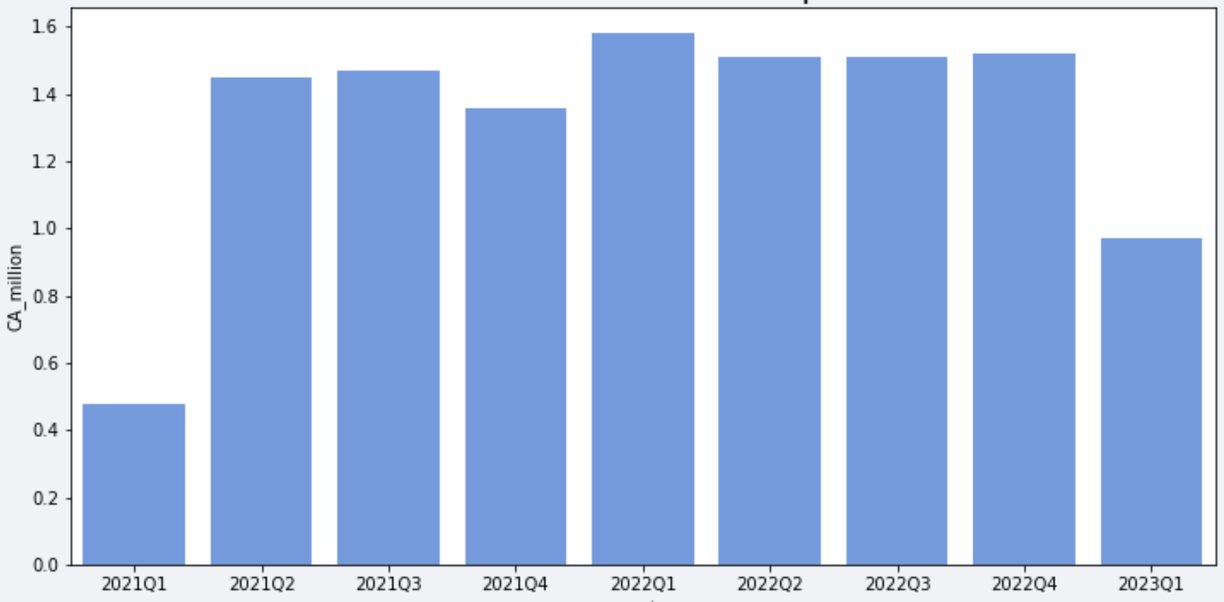


year	CA	CA_million
2021	4771847.07	4.77
2022	6110089.41	6.11
2023	974384.53	0.97

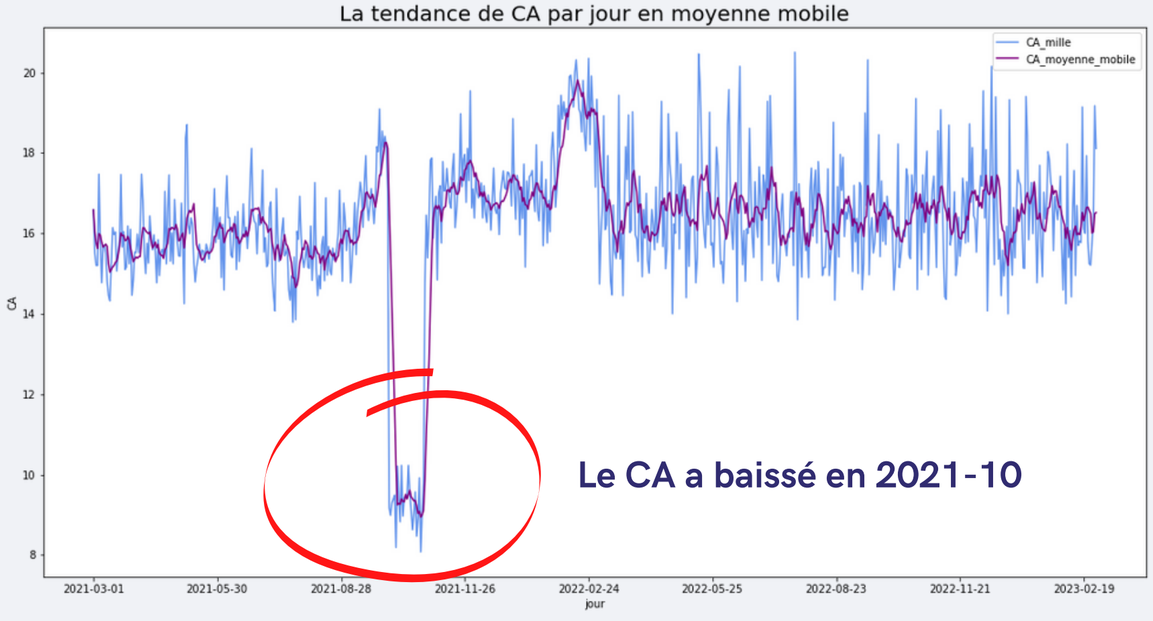
La tendance de CA par mois en moyenne mobile



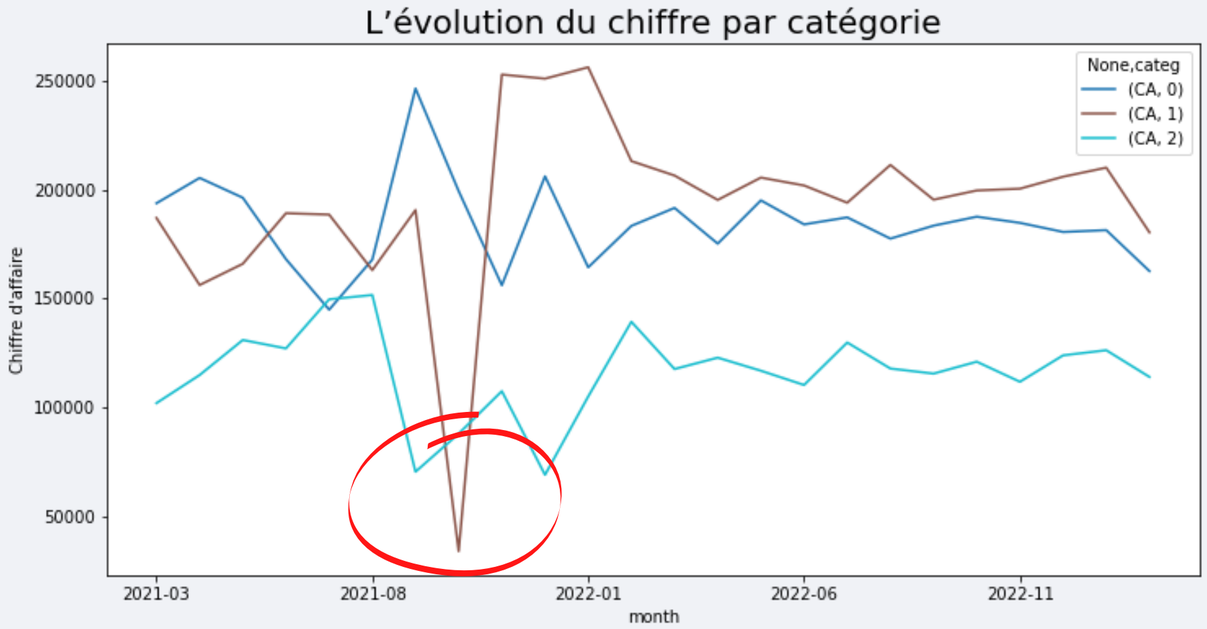
L'évolution du chiffre d'affaires par trimestre

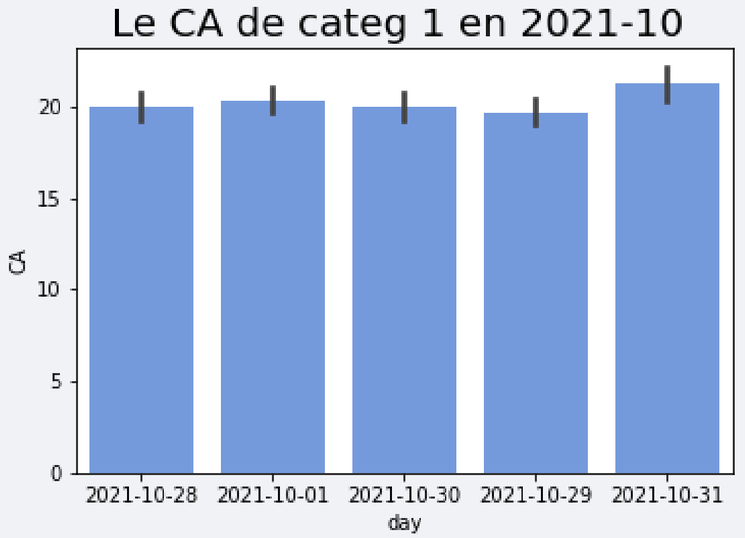


La tendance de CA par jour en moyenne mobile

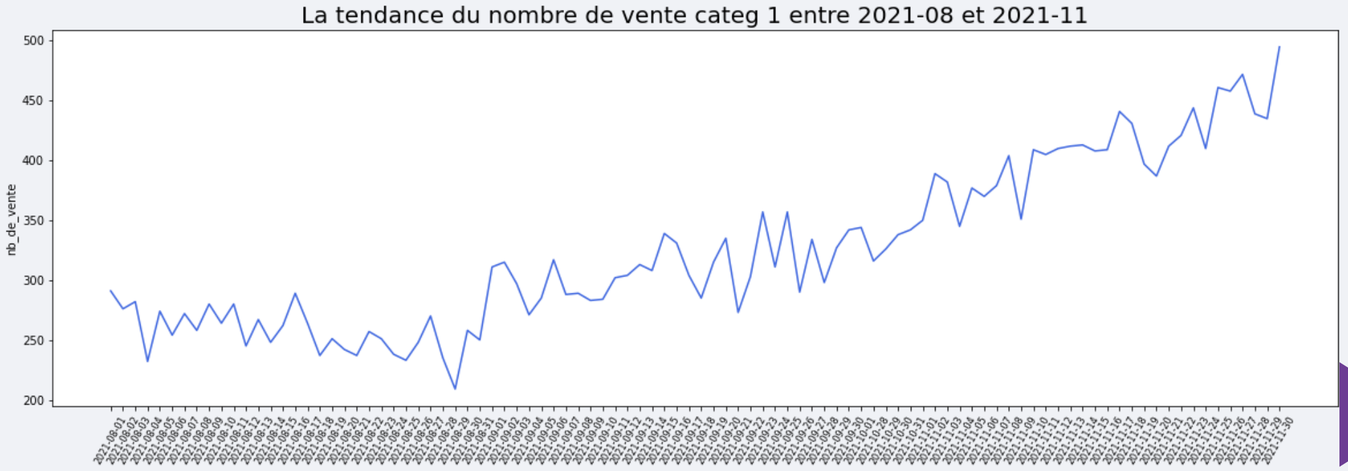


La baisse en 2021-10





***Il n'y a que 5 jours en 2021-10 qui ont de données pour la vente de categ1**



***Exclure 2021-10 pour l'analyse de tendance mais le garder pour les autres analyses**

***La tendance de categ 1 est continu avec l'absence de données pendant 02-10-2021 et 27-10-2021**

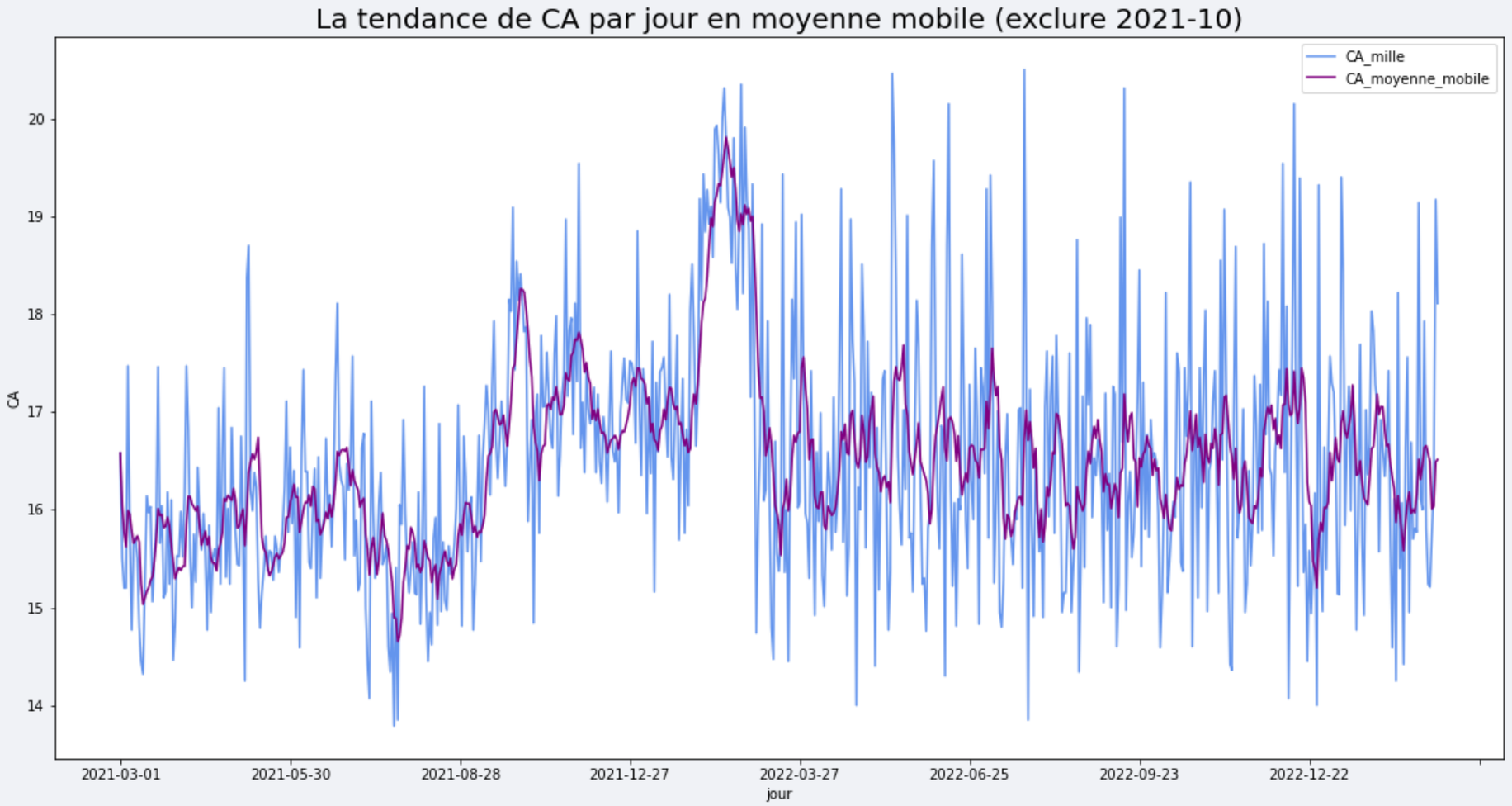
1.1.6. La tendance de CA par mois en moyenne mobile (exclure 2021-10)

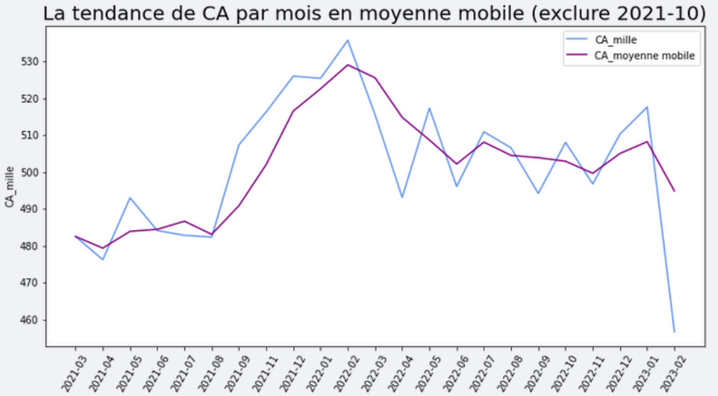
```
# exclure 2021-10
df_no10 = df.loc[(df['month'] != '2021-10'), :]

# calculer CA par mois
df_mCA_no10 = df_no10.groupby (by = 'month').sum()['CA']
df_mCA_no10 = pd.DataFrame(df_mCA_no10)
df_mCA_no10['CA_mille'] = round(df_mCA_no10['CA']/1000,2)

# calculer le moyenne mobile
df_mCA_no10['CA_moyenne mobile'] = df_mCA_no10['CA_mille'].rolling(window =3, min_periods =1).mean()
df_mCA_no10.head()
```

Exclure 2021-10 pour l'analyse de tendance



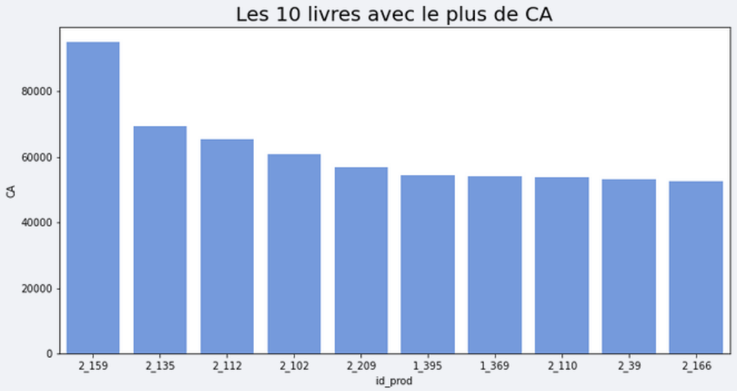


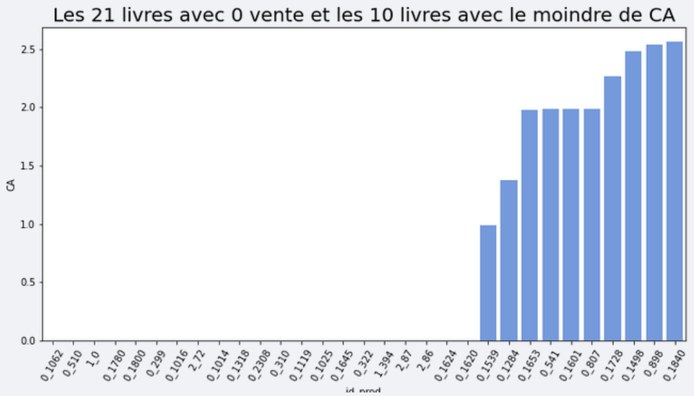
*Le CA de 2022Q1 est le plus haut.

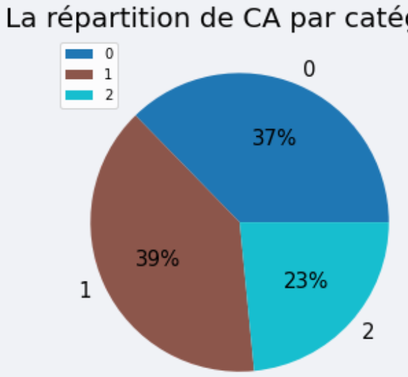
*Le CA de 2022 a augmenté par rapport à 2021.

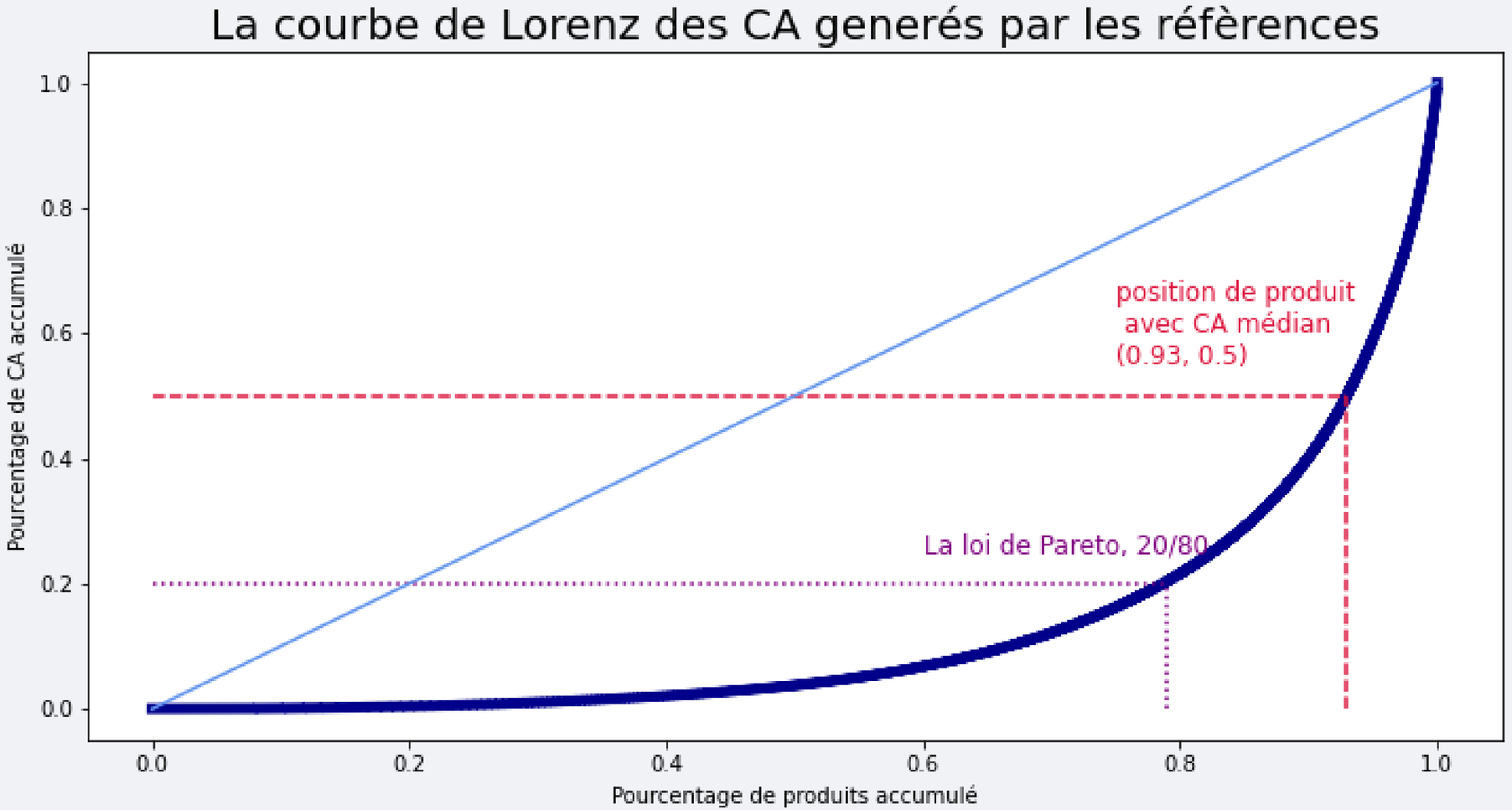
*La tendance du CA reste stable entre 15K et 18K euro par jour.

Un zoom sur les références





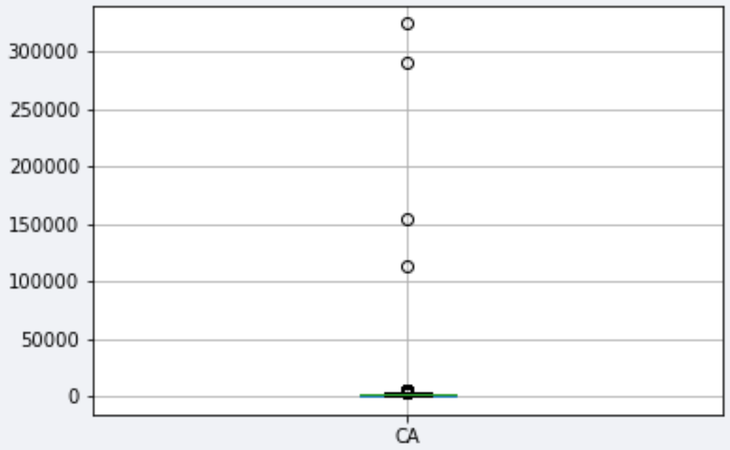




L'indice de gini (le niveau d'inégalité): 0.74

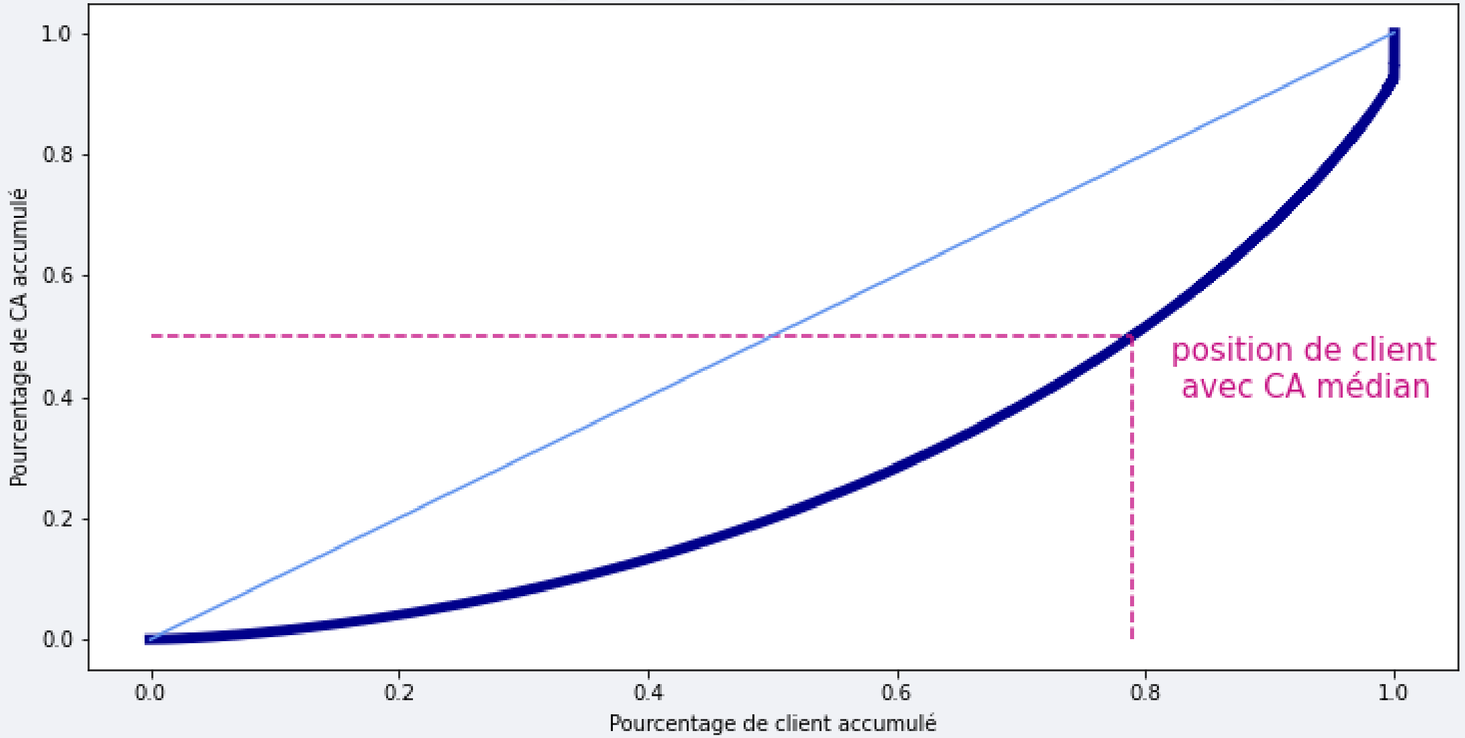
***20% des références contribuenet 80% du CA, qui refléte bien la loi de Pareto.**

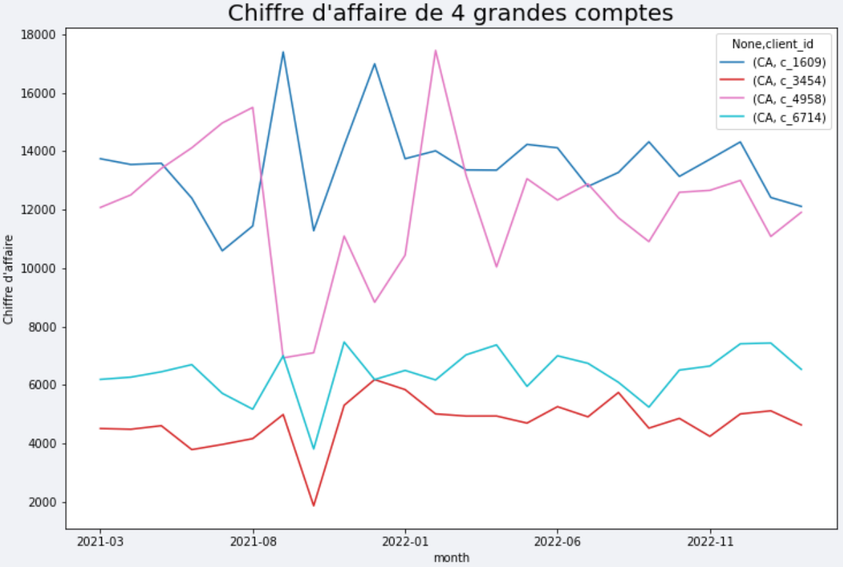
Les profils des clients: Analyser les grandes comptes



*Il y a 4 grandes comptes avec CA beaucoup plus haute que les autres. Il faut analyser les 4 grandes compte séparément.

La courbe de Lorenz sur les CA generés généré par clients(avec 4 grandes comptes)

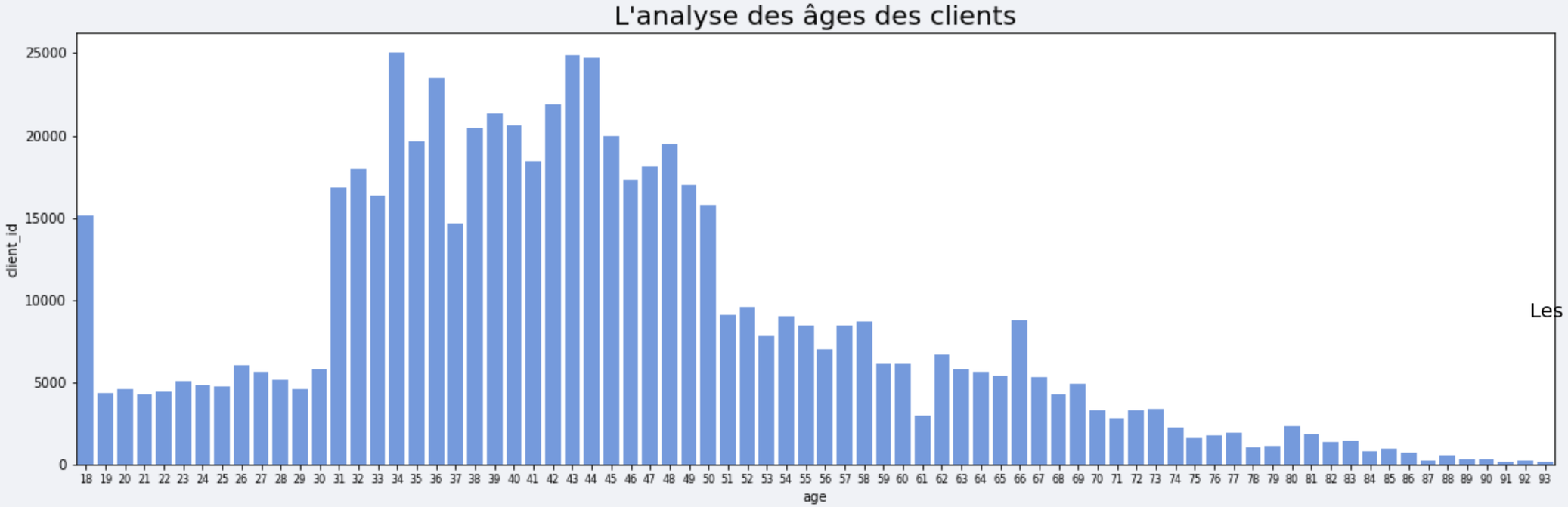


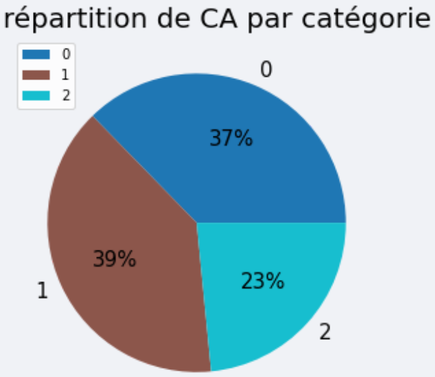


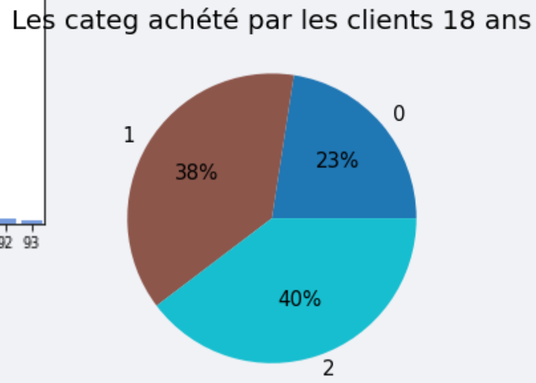
L'indice de gini (le niveau d'inégalité): 0.45

***20% de clients contribuent 50% de CA.**

Analyser les clients individuels (sans grandes comptes)

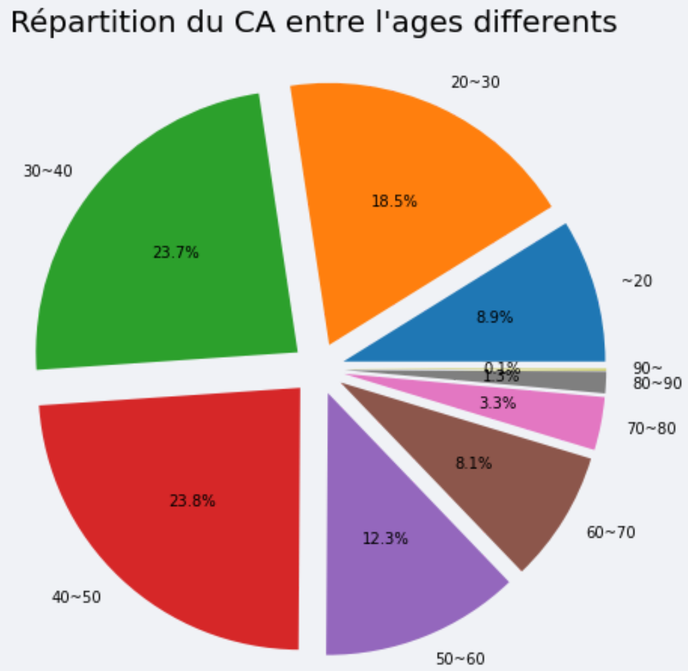


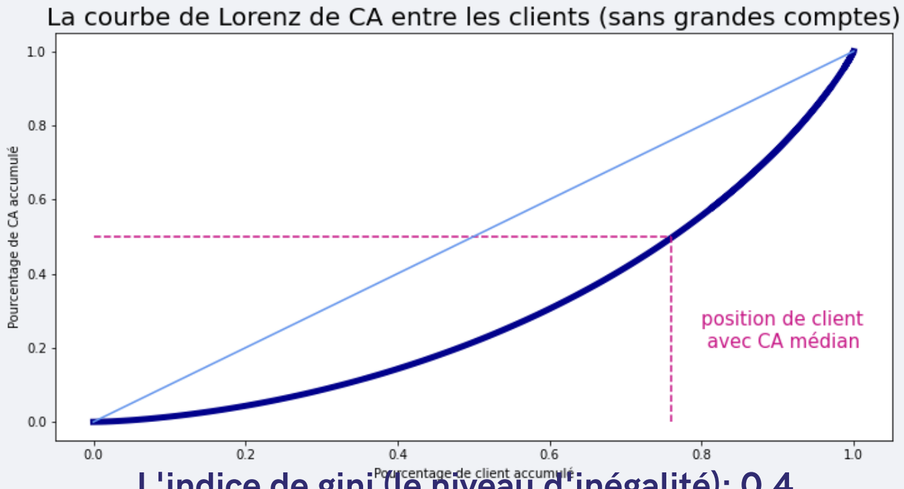




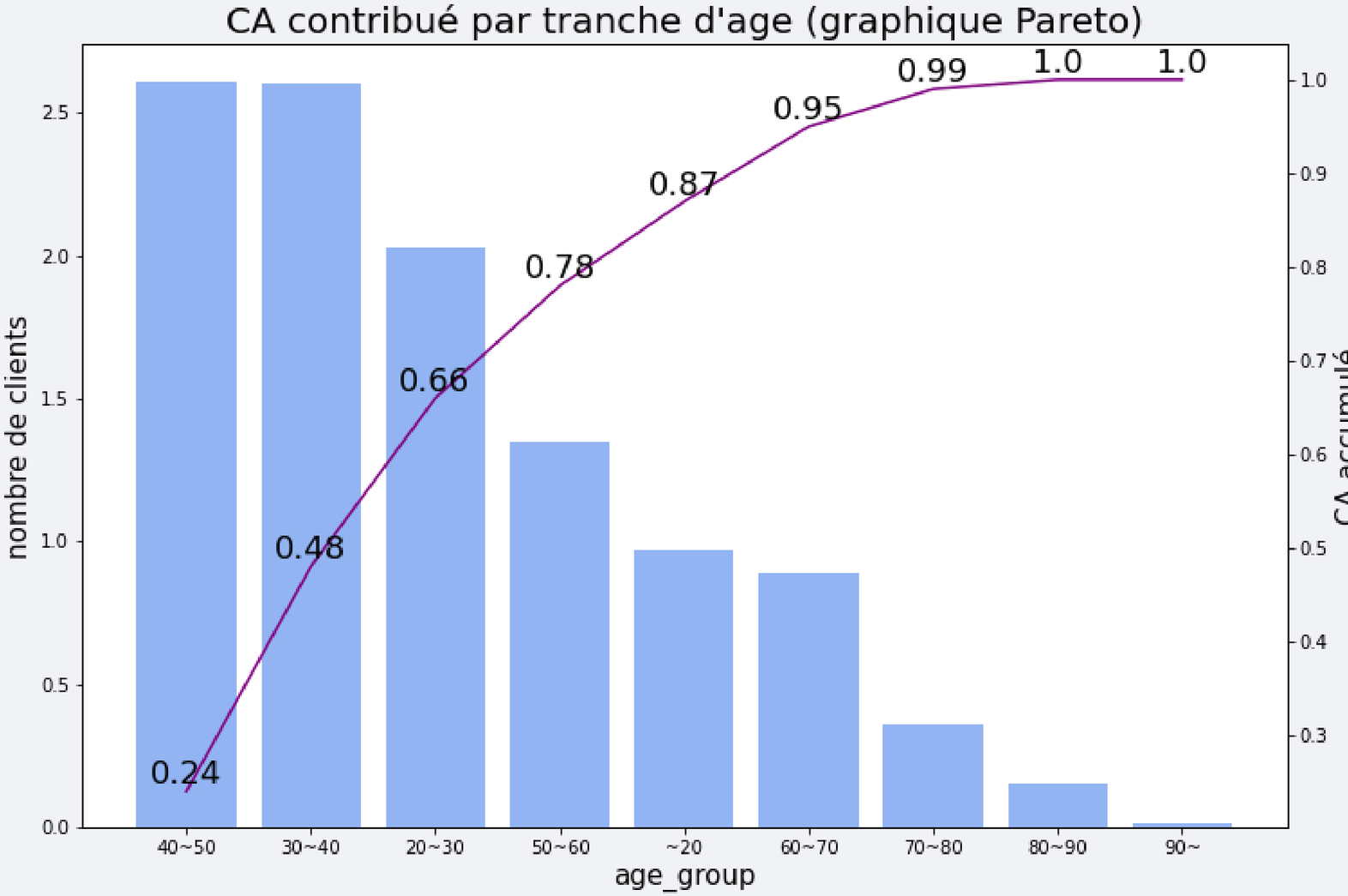
- La majorité est entre 31-50 ans.
- L'âge 18 est beacoup plus que 19, 20, qui est anormal. Il peut à cause des raisons suivant:
- 1. fausse d'information donnés par les clients 2. les étudiants pour acheter les livres de courses
  On va garder les données originals parce que l'impact d'analyse est limité.

Répartition du CA entre les ages differents (sans grandes comptes)



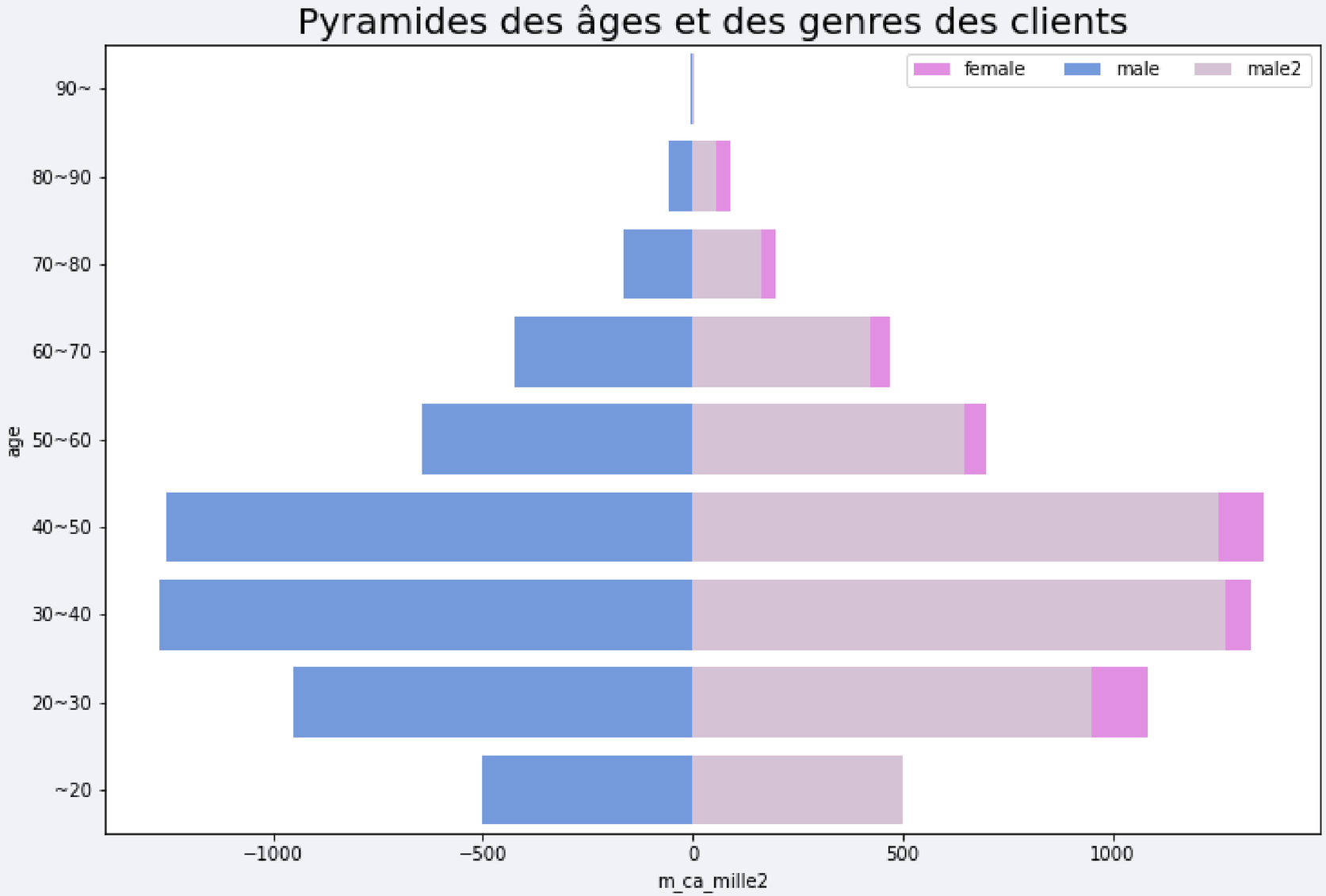


L'indice de gini (le niveau d'inégalité): 0.4



*Les groups '40-50', '30-40', '20-30', '50-60' contribuent 78% de CA.

Répartition du CA entre les genres des client



*Les formes des répartitions entre les groupes des âges sont pareils entre les sexe differents.

*Les femmes contribuent un petit peu plus que les hommes.

*Les groups '40-50', '30-40', '20-30', '50-60' contribuent le plus.

Les corrélations

01 Genres vs. catégories

02 Les âges vs. les catégories

03 Les âges vs. les montants totals des achats

04 L'âges vs. la fréquence d'achat

05 L'âges vs. la taille du panier moyen

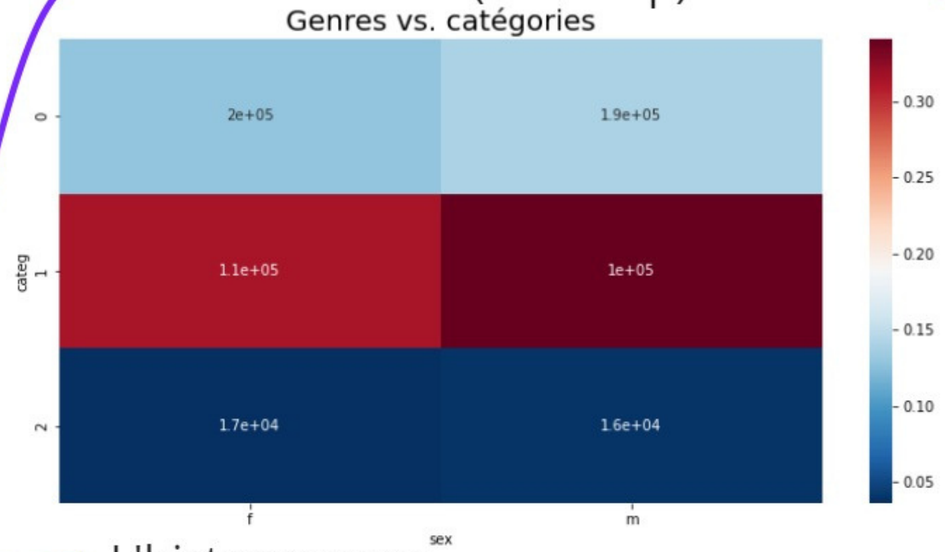
06* Jour de la semaine vs. la taille du panier

Genres vs. catégories (quali, quali)

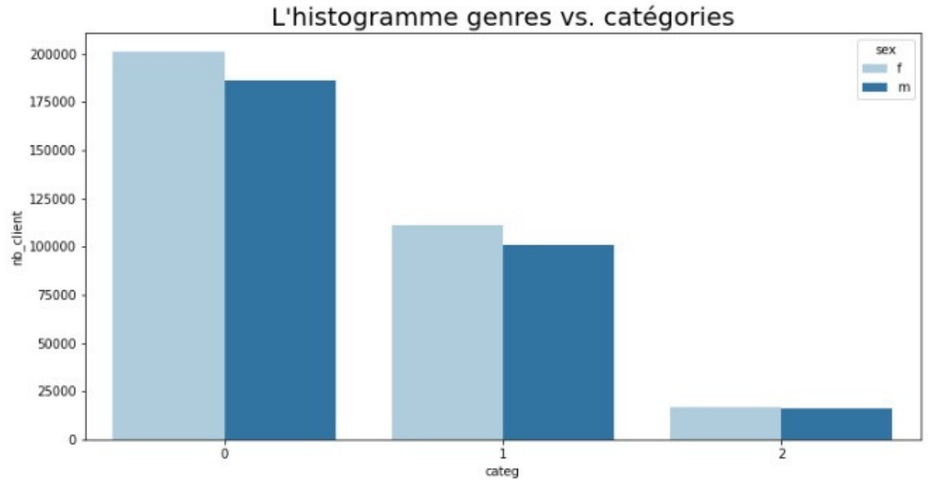
Genres vs. catégories

Quali Quali

Carte de chaleur (heatmap)



L'histogramme



Teste Chi2

P-value:
0.00255, P<0.05

H0: refuse

HA: accepte

HA: Les catégories des livres achetés sont differents entre les genres de clients.

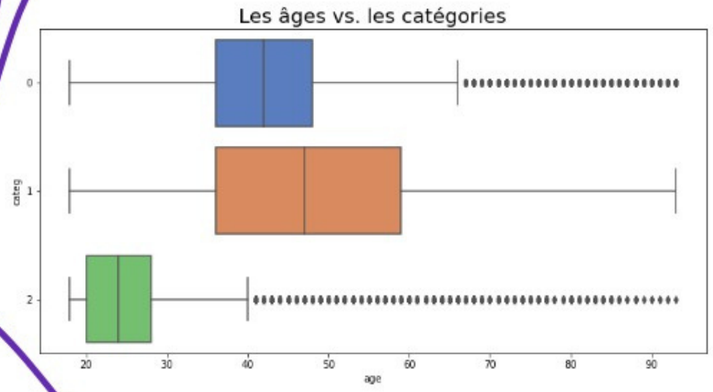
Les âges vs. les catégories (quanti, quali)

Eta2

Eta2 = 0.27, Il y a une corrélation entre les catégories des livres achetés et les âges differents des clients.

HA: Il y a une corrélation entre les catégories des livres achetés et les âges differents des clients.

Graphique boite à moustache



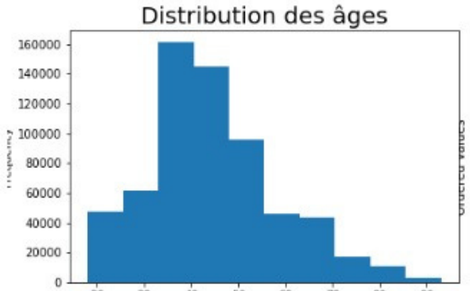
Anova — P-value < 0.05, on rejette H0

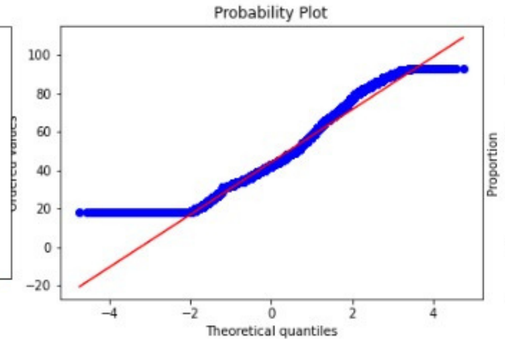
normalité validé

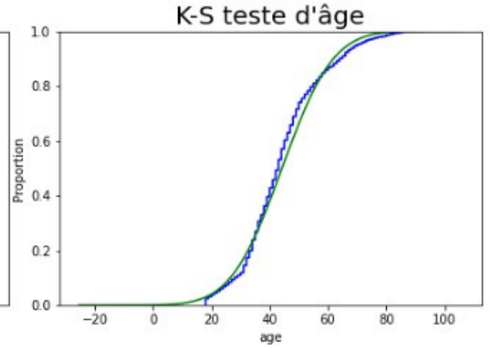
Kruskal Wallis — P-value = 0.0, on rejette H0

Teste non parametre

Teste loi normalité





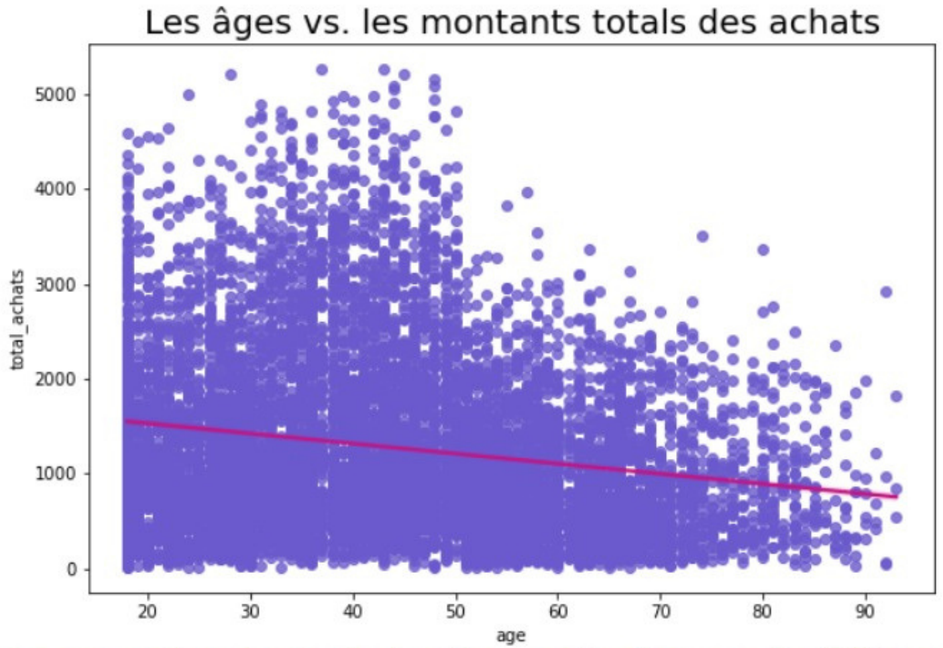


Kolmogorov-Smirnov--Teste normalité:

Basé aux les graphiques, la distribution d'âge suit bien la loi normal après 18 ans. Cependant, il n'y a pas d'âge moins de 18 ans dans les profiles de clients, qui ne correspondent pas la distribution normale. Ici, on va ultiliser ANOVA quand même et le teste Kruskal Wallis dans le même temps comme le teste non-parametre.

Les âges vs. les montants totals des achats (quanti, quanti)

Graphique nuage de points



Pearson

P-value = 0.0, P-value < 0.05,
on rejette H0 et accept **HA: Les montants totals des achats sont differents entre les âges differents des clients.**
Pearson correlation coefficient = -0.19, la force du lien linéaire est faible.

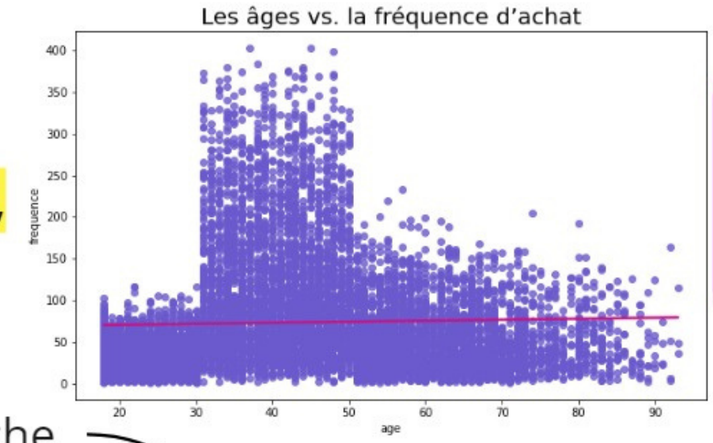
Spearman

P-value = 0.0, P-value < 0.05,
on rejette H0 et accept **HA: Les montants totals des achats sont differents entre les âges differents des clients.**
Spearman correlation coefficient = -0.19, la force du lien linéaire est faible.

L'âges vs. la fréquence d'achat (quanti, quanti)

Graphique nuage de points

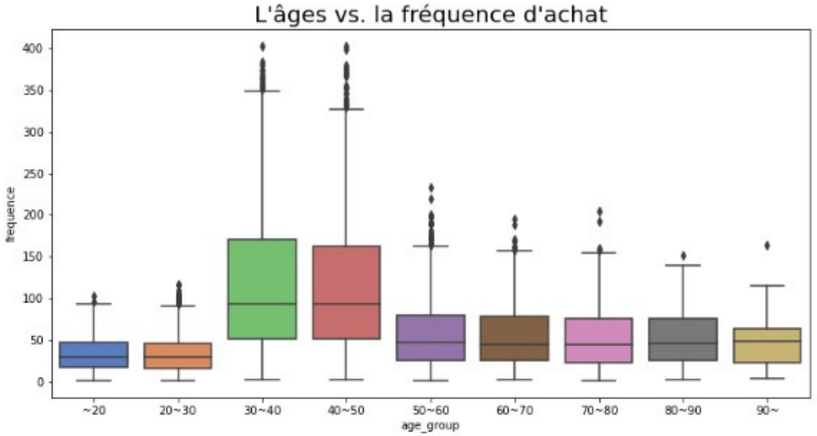
Spearman — P-value = 0.0, P-value < 0.05, on rejette H0 et accept HA



***HA: Les fréquences d'achats sont differents entre les âges differents des clients.**

Regrouper les données à quanti-quali

Graphique boite à moustache



normalité pas validé

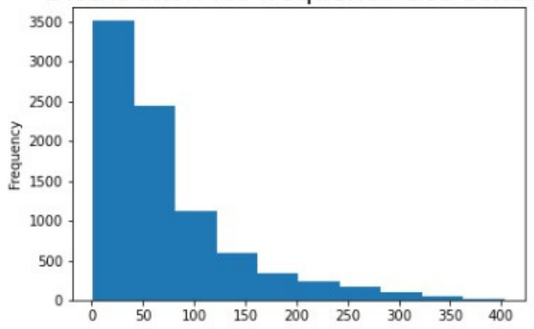
Anova

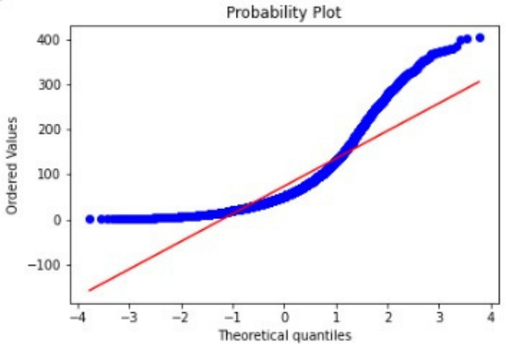
Teste loi normalité

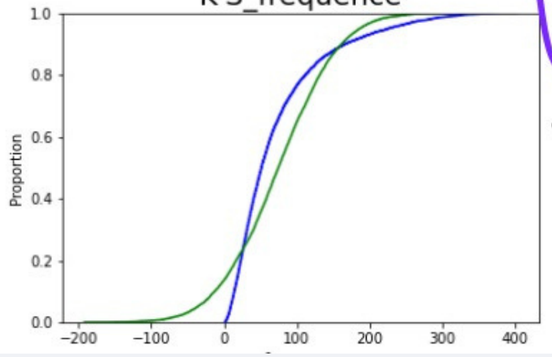
Teste non parametre

Kruskal-Wallis

Distribution les fréquence des achats







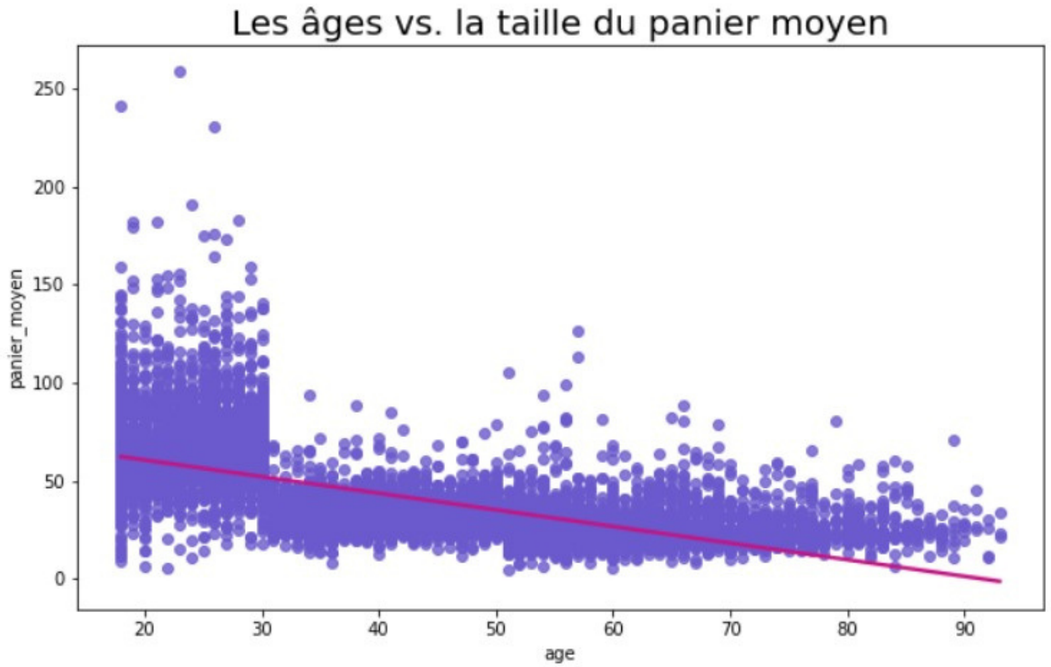
P-value = 0.0, on rejette H0.

Kolmogorov-Smirnov--Teste normalité:

La fréquence d'achat ne suit pas la distribution normale. Au lieu de teste ANOVA, on va ultiliser le teste Kruskal-Wallis.

L'âges vs. la taille du panier moyen (quanti, quanti)

Graphique nuage de points



Pearson

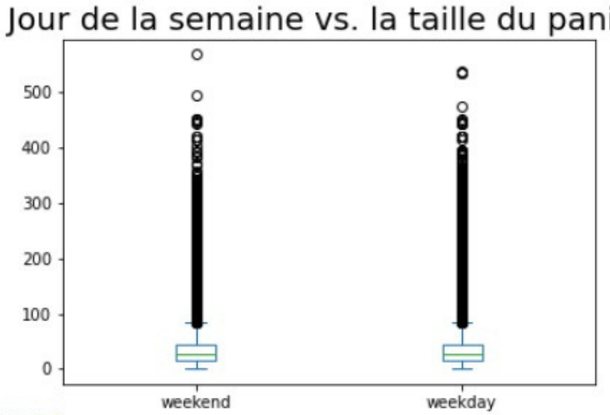
P-value = 0.0, **P-value < 0.05**, on rejette H0 et accept **HA: Les tailles des paniers moyens sont differents entre les âges differents des clients.**
Pearson correlation coefficient = -0.62, la force du lien linéaire est moyenne.

Spearman

P-value = 0.0, **P-value < 0.05**, on rejette H0 et accept **HA: Les tailles des paniers moyens sont differents entre les âges differents des clients.**
Spearman correlation coefficient = -0.7, la force du lien linéaire est moyenne.

Jour de la semaine vs. la taille du panier (quali, quanti)

Graphique boite à moustache



Teste loi normalité

T-Teste

normalité pas validé

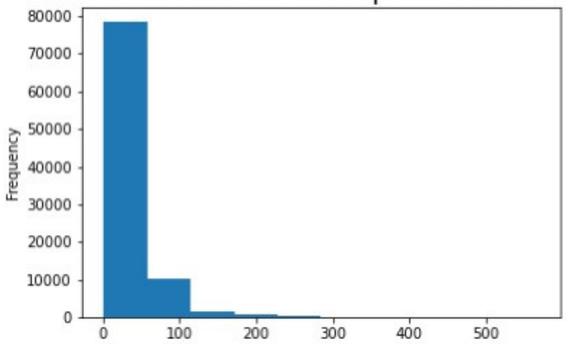
Mann-Whitney

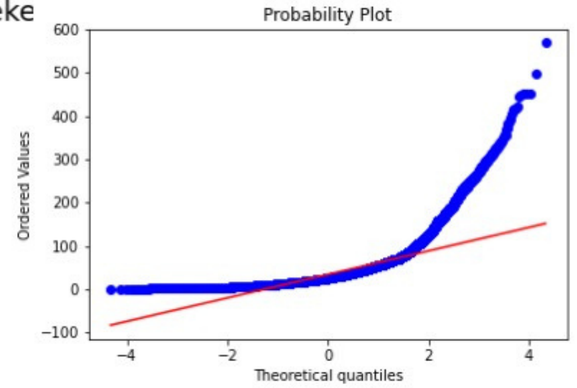
Teste non parametre

P-value > 0.05,
on accepte H0

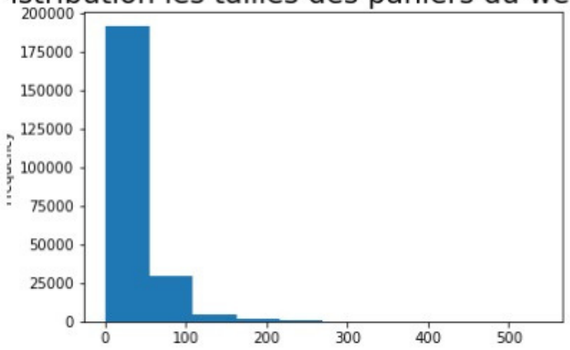
H0: Les tailles des paniers sont pareils entre weekend et les autres jours de semaines.

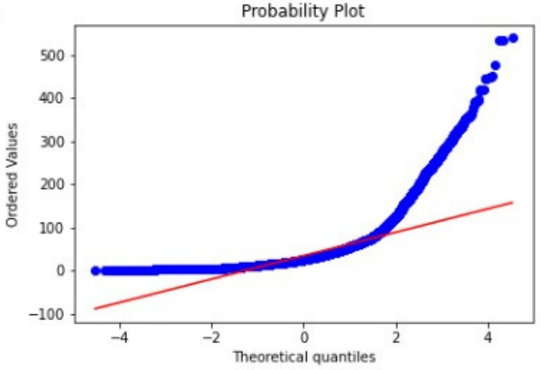
istribution des tailles des paniers du weeke





istribution les tailles des paniers du weekd





*Les tailles des paniers ne suite pas la normalité, **on ne peut pas ultiliser le T-test(Test student).**

Merci !