

e - Librairie

Analyse des Ventes sur l'année

Introduction

Cette étude se propose d'analyser les ventes d'une librairie de ventes en ligne.

Elle a été réalisée :

- Dans le cadre de la **formation DATA ANALYST** d'OpenClassRooms
- À partir d'un jeu de **données** fictif
- Avec l'aide/support de **Mr Quentin Desrousseaux** (Mentor Openclassrooms)

Choix des graphes & couleurs :

L'idée était de montrer les différentes fonctionnalités proposées par les librairies «matplotlib» et «seaborn».

→ Pas de suivi dans les couleurs utilisées (catégories par ex.)

Sources Web

Lien vers jeu de données



Institut national de la statistique et des études économiques

Mesurer pour comprendre



WIKIPEDIA
The Free Encyclopedia

Sommaire

- Introduction (remerciements)
- M1 - Nettoyage des Données 
 - Table des Clients
 - Table des Produits
 - Table des Transactions
 - Constitution d'un Dataframe « enrichi » des ventes
- M2 - Analyses & Graphes 
 - CLIENTS
 - Age / Tranches & répartition par Sexe
 - Les 10 plus « gros » clients (CA Annuel)
 - PRODUITS
 - Distribution des Tarifs Produits par Catégorie
 - Les 10 Produits les plus vendus (Nb Ventes & CA)
 - TRANSACTIONS / VENTES
 - Diverses Etudes CA Ventes & Nb de Ventes / Mois
 - Courbe de Lorenz / Indice Gini
(Distribution des Ventes annuelles)
- M3 - Corrélations (Analyses Bivariées) 
 - Sexe des Clients / Catégories de Produits
 - Age des Clients & : Montant Total Achats
Fréquence Achats Mensuelle
Taille Panier Moyen
Catégories de Produits
- Questions / Contact 



Nettoyage des Données



Outils de mesures de tendance centrale & et de dispersion



SGBD : Opérations sur les tables, requêtage (sélection, jointures, intégrité référentielles)

Le nettoyage des données a été réalisé avec les deux solutions ci-dessus, chacune présentant avantages et inconvénients.

Dans cette présentation, seul le Data Cleaning avec Python et la librairie Pandas sera évoqué.

Liste des Actions réalisées :

- **Clé Primaire (PK)** → Choix / Proposition
Recherche Doublons (PK)
- **Recherche Valeurs nulles** et mise à jour (+ suppression/remplacement)
PK & autres colonnes
- **Recherche Valeurs aberrantes** résiduelles (+ suppression/remplacement)
Ensemble des colonnes
- **Contrôles finaux** de la table (Informations chiffrées : totaux, min, max ...)
- **Contrôle d'intégrité** entre Tables (sur PK) (Jointures entre les tables sur PK)
Ex : Le client de la table des ventes doit exister dans la table des clients.

Nettoyage des Données

Table des Clients

Table "customers.csv" - DataFrame "cst"			
ZONE	LIBELLE	TYPE	
client_id	Code Client	CHAR	
sex	Sexe du Client	INT	
birth	Année Naissance Client	INT	
age	Age du client	INT	

client_id	sex	birth	age
c_4410	f	1967	55
c_7839	f	1975	47
c_1699	f	1984	38
c_5961	f	1962	60
c_5320	m	1943	79

- Choix Clé Primaire → **client_id**
- Recherche de Valeurs Nulles
 - suppression sur la clé primaire
 - init des autres colonnes
- Recherche de Doublons
 - sur la clé primaire
- Recherche de Valeurs aberrantes
 - valeurs distinctes zone « sex »
 - Min & Max Année naissance
 - (contrôle sur RECAP)
- Ajout d'une nouvelle zone « age »

```

1 cst.dropna(subset=['client_id'], inplace=True)
2 cst.sex = cst.sex.fillna('')
3 cst.birth = cst.birth.fillna(0)

1 cst.isnull().values.any()
False

```

```

1 cst[cst.duplicated(subset=['client_id']) == True].head()
2 #cst.drop_duplicates(subset='client_id', inplace=True)

client_id  sex  birth  age

```

```

1 # Valeurs distinctes de la colonne "sex"
2 cst.sex.unique()

array(['f', 'm'], dtype=object)

```

```
1 cst['age'] = (2022 - cst['birth']).astype('int')
```

RECAP Final

```

1 cst.info()
2 cst.describe(include="all")

<class 'pandas.core.frame.DataFrame'>
Int64Index: 8623 entries, 0 to 8622
Data columns (total 4 columns):
client_id    8623 non-null object
sex          8623 non-null object
birth        8623 non-null int64
age          8623 non-null int32
dtypes: int32(1), int64(1), object(2)
memory usage: 303.2+ KB

```

	client_id	sex	birth	age
count	8623	8623	8,623.00	8,623.00
unique	8623	2	nan	nan
top	c_2289	f	nan	nan
freq	1	4491	nan	nan
mean	NaN	NaN	1,978.28	43.72
std	NaN	NaN	16.92	16.92
min	NaN	NaN	1,929.00	18.00
25%	NaN	NaN	1,966.00	30.00
50%	NaN	NaN	1,979.00	43.00
75%	NaN	NaN	1,992.00	56.00
max	NaN	NaN	2,004.00	93.00

Nettoyage des Données

Table des Produits

Table products.csv" - DataFrame "prd"		
ZONE	LIBELLE	TYPE
id_prod	Code Produit	CHAR
price	Tarif du produit	INT
categ	Catégorie du produit	INT

id_prod	price	categ
0_1421	19.99	0
0_1368	5.13	0
0_731	17.99	0
1_587	4.99	1
0_1507	3.99	0

- Choix Clé Primaire → **id_prod / (categ)**
- Recherche de Valeurs Nulles
 - suppression sur la clé primaire
 - init des autres colonnes
- Recherche de Doublons
 - sur la clé primaire
- Recherche de Valeurs aberrantes
 - valeurs distinctes zone « categ »
 - produits ayant un prix négatif & suppression éventuelle

```

1 prd.dropna(subset=['categ', 'id_prod'], inplace=True)
2 prd.price = prd.price.fillna(0)
3 prd.categ = prd.categ.fillna(0)

1 prd.isnull().values.any()
False

```

```

1 prd[prd.duplicated(subset=['categ', 'id_prod']) == True].head()
2 # prd.drop_duplicates(subset=['categ', 'id_prod'], inplace=True)

id_prod  price  categ

```

- valeurs distinctes zone « categ »
- produits ayant un prix négatif & suppression éventuelle

```

1 # Valeurs distinctes de la colonne "categorie" et Leurs quantité respectives
2 prd.groupby('categ')['id_prod'].count().reset_index()

```

```

1 # Recherche de produits ayant un prix négatif
2 prd[prd.price < 0].head()

id_prod  price  categ

```

731	T_0	-1.00	0
-----	-----	-------	---

```

1 # Suppression des Lignes dont Le Produit a un prix négatif
2 prd.drop(prd[prd.price < 0].index, inplace=True)

```

RECAP Final		
id_prod	price	categ
count	3286	3,286.00
unique	3286	nan
top	0_265	nan
freq	1	nan
mean	NaN	21.86
std	NaN	29.85
min	NaN	0.62
25%	NaN	6.99
50%	NaN	13.07
75%	NaN	22.99
max	NaN	300.00

categ	id_prod
0	2309
1	739
2	239



Nettoyage des Données

Table des Transactions

Table "transactions.csv" - DataFrame "tra"		
ZONE	LIBELLE	TYPE
id_prod	Code Produit	CHAR
date	Date-Heure de la transaction	INT
session_id	Code de Session connexion	CHAR
client_id	Code Client	CHAR

id_prod	date	session_id	client_id
0_1483	2021-04-10 18:37:28.723910	s_18746	c_4450
2_226	2022-02-03 01:55:53.276402	s_159142	c_277
1_374	2021-09-23 15:13:46.938559	s_94290	c_4270
0_2186	2021-10-17 03:27:18.783634	s_105936	c_4597
0_1351	2021-07-17 20:34:25.800563	s_63642	c_1242

- Choix Clé Primaire → session_id / date (couple)

- Recherche de Valeurs Nulles
 - suppression sur la clé primaire
 - init des autres colonnes

```

1 tra.dropna(subset=['session_id', 'date'], inplace=True)
2 tra.id_prod = tra.id_prod.fillna('')
3 tra.client_id = tra.client_id.fillna(0).prd.isnull().values.any()

1 tra.isnull().values.any()
False

```

- Recherche de Doublons & suppression
 - sur la clé primaire

1 tra[tra.duplicated(subset=['session_id', 'date']) == True].head()				
id_prod		date	session_id	client_id
27161	T_0	test_2021-03-01 02:30:02.237434	s_0	ct_0
34387	T_0	test_2021-03-01 02:30:02.237443	s_0	ct_0
48425	T_0	test_2021-03-01 02:30:02.237443	s_0	ct_1

1 tra[tra.date.str.contains("test_")].head()				
id_prod		date	session_id	client_id
1431	T_0	test_2021-03-01 02:30:02.237420	s_0	ct_1
2365	T_0	test_2021-03-01 02:30:02.237446	s_0	ct_1
2895	T_0	test_2021-03-01 02:30:02.237414	s_0	ct_1

```

1 # Suppression des valeurs date contenant "test_"
2 tra.drop(tra[tra.date.str.contains("test_") == True].index, inplace=True)

```

Nettoyage des Données

Table des Transactions (suite)

○ Recherche de Valeurs aberrantes

Utilisation des jointures pour ne conserver que lignes de transactions dont :

- le client existe dans la table client
- le produit existe dans la table produit
- cohérence du nb de Ventes mensuelles par catégorie

```
1 tra = pd.merge(tra, cst, how='outer')
2 tra = pd.merge(tra, prd, how='outer')
3 tra = tra.dropna()
```

```
1 tra.isnull().values.any()
```

```
False
```

Trou de données manquantes entre le 2 et 27 Octobre inclus pour la catégorie «1».

```
1 df = sal.groupby(['year', 'month', 'day', 'categ'])['prod'].count().reset_index(name="counts")
2 df = df[(df['categ']==1)&(df['month']==10)].sort_values(['categ', 'year', 'month', 'day'])
3 df.head()
```

	year	month	day	categ	counts
643	2021	10	1	1	344
698	2021	10	28	1	316
701	2021	10	29	1	326
704	2021	10	30	1	338
707	2021	10	31	1	342

Probablement un Problème Technique survenu sur le site.

session_id	client	sex	age	birthyear	prod	price	categ	period	year	month	day	time	session_date
s_98332	c_5765	m	70	1952	1_469	5.99	1	2021-10	2021	10	1	08:34	2021-10-01 08:34:35.747969
s_98343	c_7902	m	59	1963	1_204	6.73	1	2021-10	2021	10	1	09:08	2021-10-01 09:08:50.519777
s_98549	c_4073	f	37	1985	1_382	61.52	1	2021-10	2021	10	1	18:24	2021-10-01 18:24:35.034271
s_98300	c_1006	m	43	1979	1_156	32.67	1	2021-10	2021	10	1	06:41	2021-10-01 06:41:57.331531
s_111252	c_4096	m	60	1962	1_596	11.12	1	2021-10	2021	10	28	08:10	2021-10-28 08:10:52.761426
s_111122	c_7146	m	51	1971	1_278	19.18	1	2021-10	2021	10	28	01:06	2021-10-28 01:06:40.873286
s_111229	c_2418	f	35	1987	1_278	19.18	1	2021-10	2021	10	28	06:49	2021-10-28 06:49:33.121261
s_111147	c_6680	m	43	1979	1_407	15.99	1	2021-10	2021	10	28	02:31	2021-10-28 02:31:18.401229



Nettoyage des Données

Constitution d'un nouveau Dataframe « sal » enrichi des Ventes.

A partir du Dataframe « tra » précédemment consolidé, création de nouvelles colonnes (en gris ci-contre), mise en forme et réordonnancement.

```

1 sal = tra.copy()
2 sal.columns = ['prod', 'session_date', 'session_id', 'client', 'sex', 'birthyear', 'age', 'price', 'categ']
3 sal['age'] = (2022 - sal['birthyear']).astype('int')
4 sal['birthyear'] = sal['birthyear'].astype('int')
5 sal['categ'] = sal['categ'].astype('int')
6 sal['period'] = sal['session_date'].str[0:7:1]
7 sal['year'] = sal['session_date'].str[0:4:1].astype('int')
8 sal['month'] = sal['session_date'].str[5:7:1].astype('int')
9 sal['day'] = sal['session_date'].str[8:10:1].astype('int')
10 sal['time'] = sal['session_date'].str[11:16:1]
11
12 cols = ['session_id', 'client', 'sex', 'age', 'birthyear', 'prod', 'price', 'categ', 'period',
13 'year', 'month', 'day', 'time', 'session_date']
14 sal = sal[cols]
```

session_id	client	sex	age	birthyear	prod	price	categ	period	year	month	day	time	session_date
s_18746	c_4450	f	45	1977	0_1483	4.99	0	2021-04	2021	4	10	18:37	2021-04-10 18:37:28.723910
s_140787	c_5433	f	41	1981	0_1483	4.99	0	2021-12	2021	12	27	11:11	2021-12-27 11:11:12.123067
s_110736	c_857	m	37	1985	0_1483	4.99	0	2021-10	2021	10	27	04:56	2021-10-27 04:56:38.293970
s_57626	c_3679	f	33	1989	0_1483	4.99	0	2021-07	2021	7	4	06:43	2021-07-04 06:43:45.676567
s_92165	c_1609	m	42	1980	0_1483	4.99	0	2021-09	2021	9	19	08:45	2021-09-19 08:45:43.735331

Creation Dataframe "sal" - Ventes		
ZONE	LIBELLE	TYPE
session_id	Code de Session connexion	CHAR
client	Code Client (de la Vente)	CHAR
sex	Sexe du Client (de la Vente)	CHAR
age	Age du Client (de la Vente)	INT
birthyear	Année Naissance Client (de la Vente)	INT
prod	Code Produit (de la Vente)	CHAR
price	Tarif du produit (de la Vente)	FLOAT
categ	Catégorie du Produit (de la Vente)	INT
period	Periode YYYY-MM (de la Vente)	CHAR
year	Année (de la Vente)	INT
month	Mois (de la Vente)	INT
day	Jour (de la Vente)	INT
time	Heure (de la Vente)	CHAR
session_date	Date-Heure (de la Vente) Originale	CHAR

Export des Dataframes « consolidés » pour Analyses & Graphes

EXPORT des Dataframes consolidés : dans des fichiers "csv"

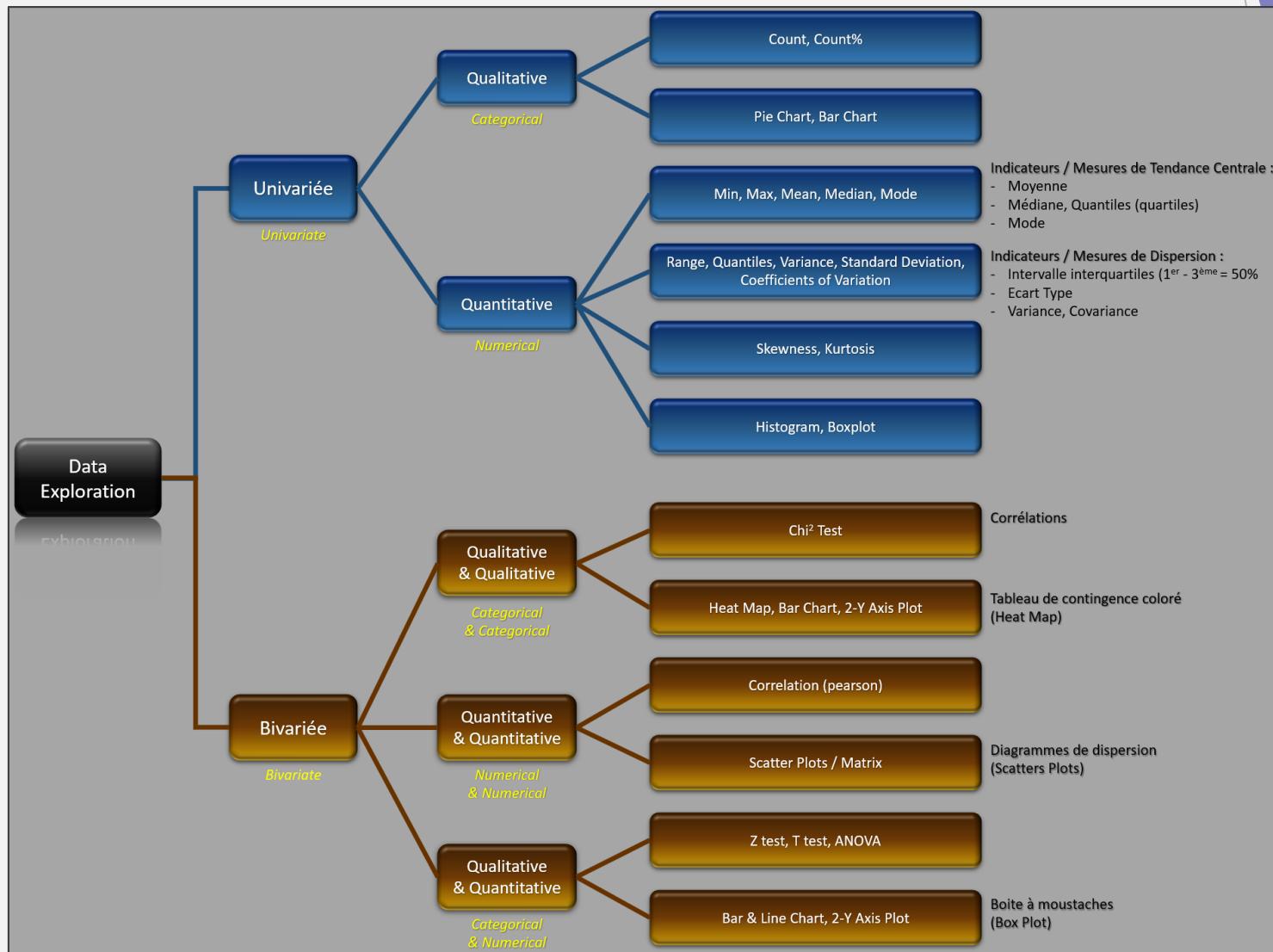
```

1 sal.to_csv('DATA/sales.csv', sep=',', encoding='utf-8', index=False)
2 cst.to_csv('DATA/cst.csv', sep=',', encoding='utf-8', index=False)
3 prd.to_csv('DATA/prd.csv', sep=',', encoding='utf-8', index=False)
```



Analyses & Graphes

Data Exploration - « Mind Map »

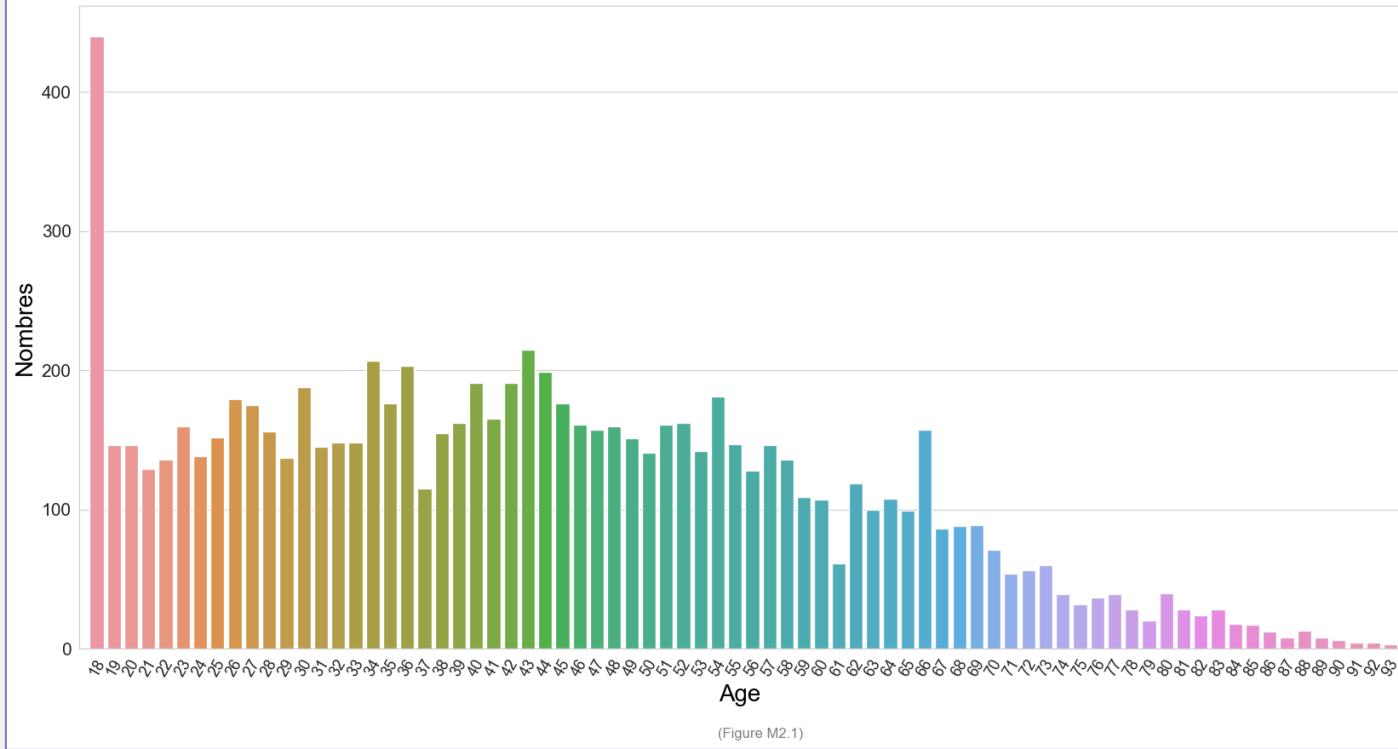


Analyses & Graphes

CLIENTS

Ages des clients

Répartition Age des Clients



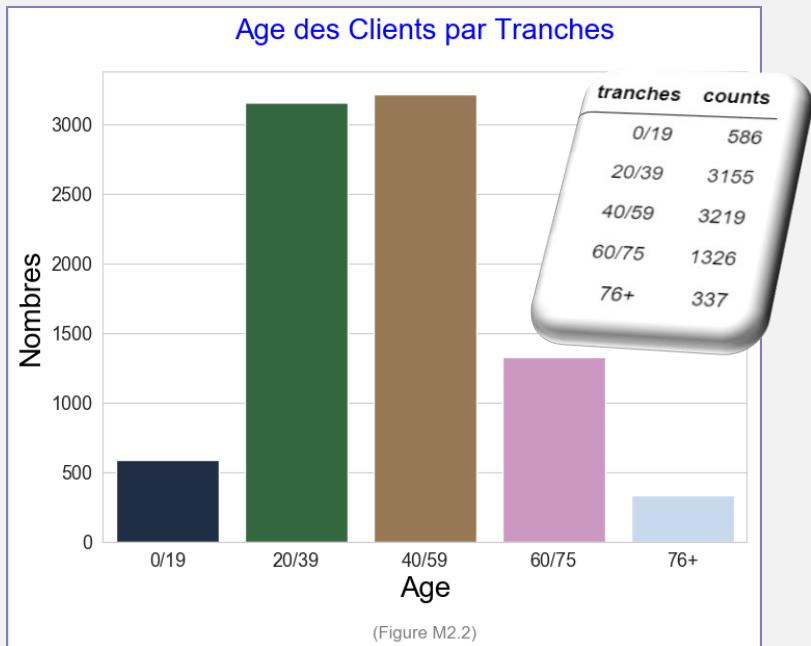
(Figure M2.1)

- *Pic exceptionnel pour les « 18 ans » ! A interpréter comme une tranche d'âge « < 18ans »*
- *Distribution assez homogène et crédible pour un échantillon de population de personnes.
- Le Nombre de Clients décroît à mesure que l'âge augmente.*

Analyses & Graphes

CLIENTS

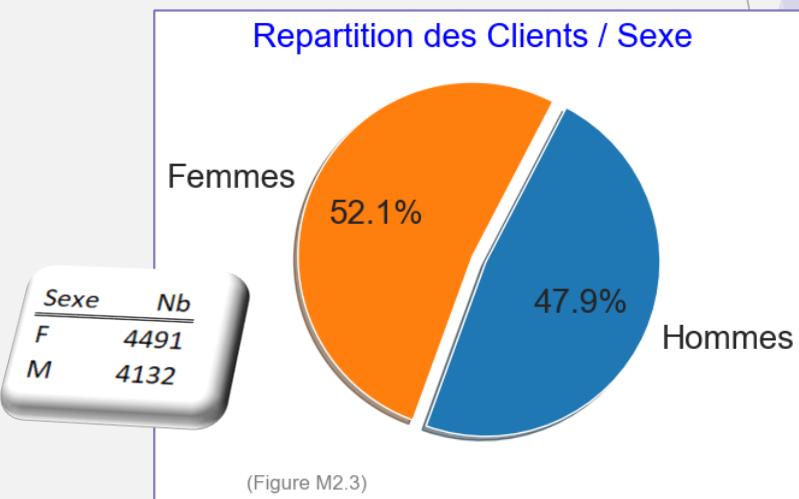
Répartition des Ages par Tranches & répartition par Sexe



- Les tranches de population « active » entre 20-59ans représentent **la grande majorité** des clients

→ 74% des clients

Meilleur Pouvoir d'achats ?



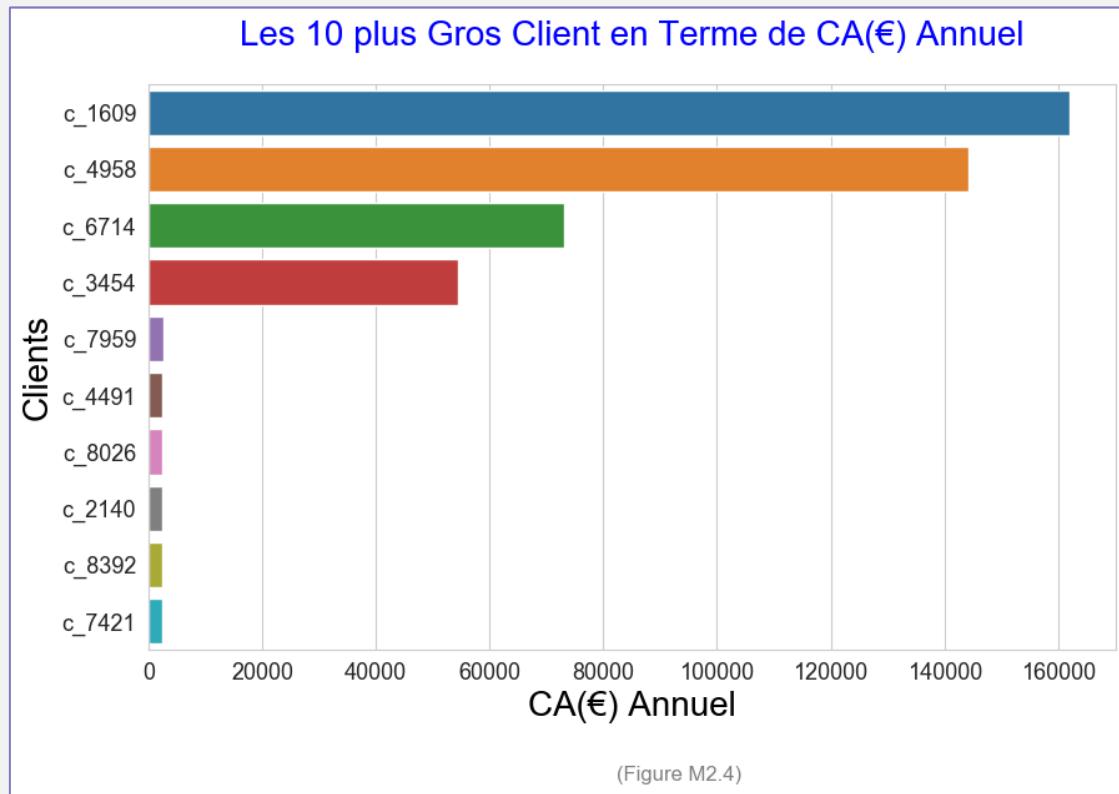
Conforme à la répartition de l'Insee
→ 51,7 % Femmes

Groupe d'âges	Femmes	Hommes	Total
Total	34 598 168	32 394 531	66 992 699

Analyses & Graphes

CLIENTS

Les 10 les plus « gros ».



- 4 très gros clients sont à noter
- Leur CA annuel s'élève à plus de 50'000 €

Ce ne sont probablement pas des "particuliers" mais plutôt des entreprises (ou institutions).

Client	CA (€)
c_1609	162'007
c_4958	144'257
c_6714	73'197
c_3454	54'443

Analyses & Graphes

PRODUITS

Distribution des Tarifs Produits par Catégorie

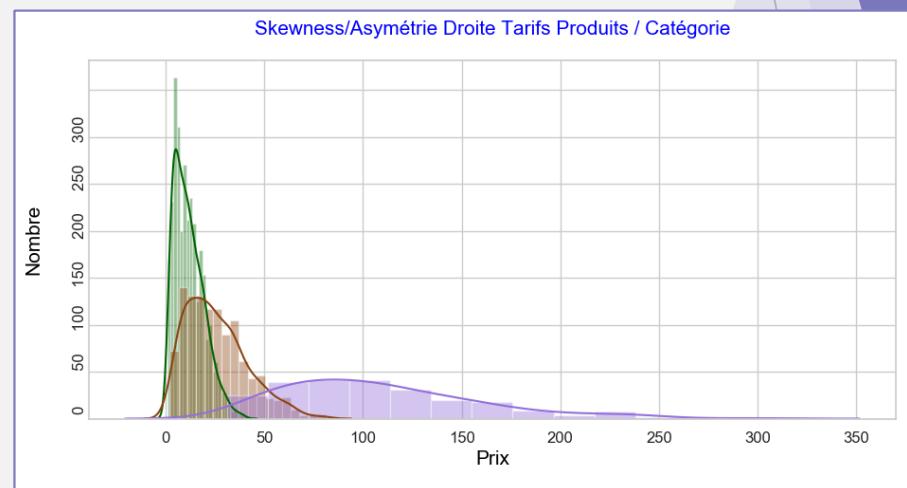
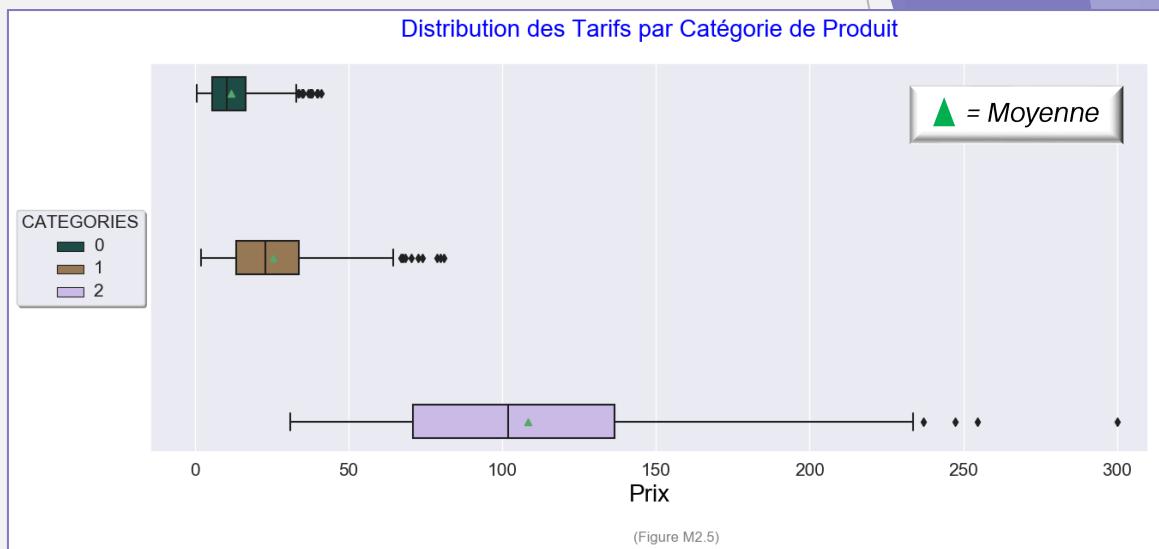


Pour chaque catégorie

Moyenne > Médiane

Caractéristique d'une distribution asymétrique à droite (skewness droit)

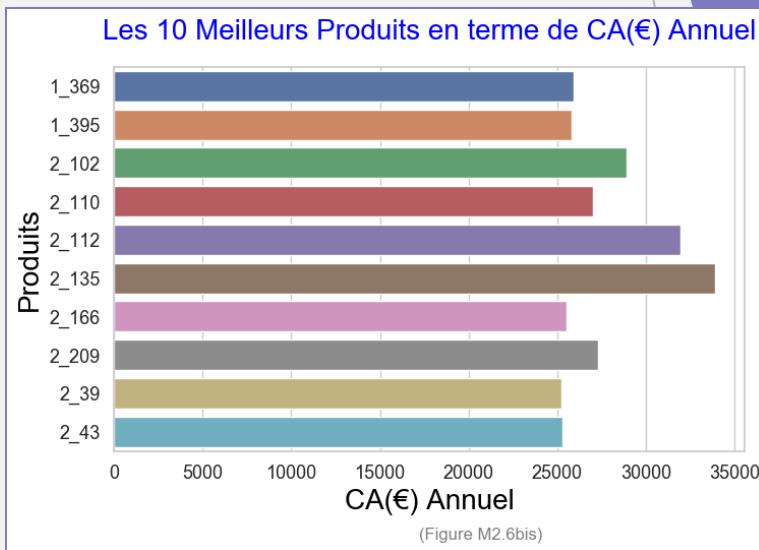
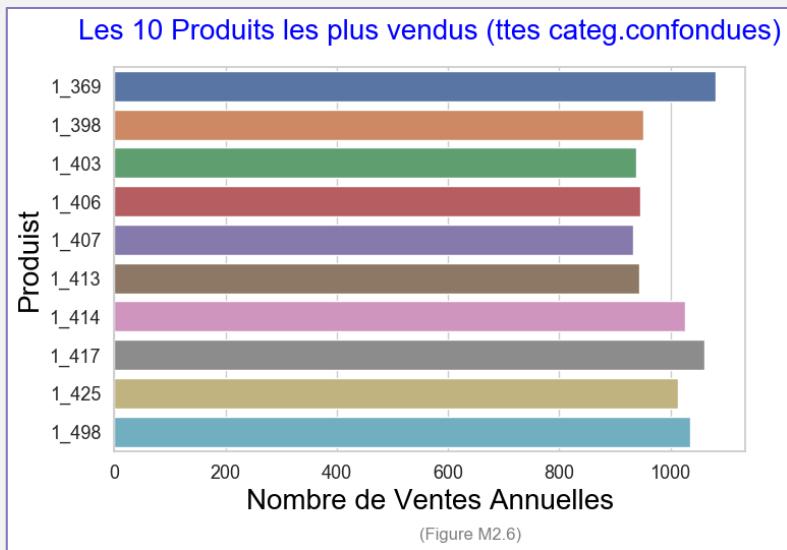
- **Catégorie 0** Concentration forte Fourchette [0-40€]
- **Catégorie 1** Concentration assez forte Fourchette [2-80€]
- **Catégorie 2** Assez large répartition Fourchette [31-300€]



Analyses & Graphes

PRODUITS

Les 10 produits les plus vendus (Nb et CA) et leur catégorie



- Tous issus de la catégorie « 1 » (intermédiaire)
- Principalement issus de la catégorie « 2 » (luxe)

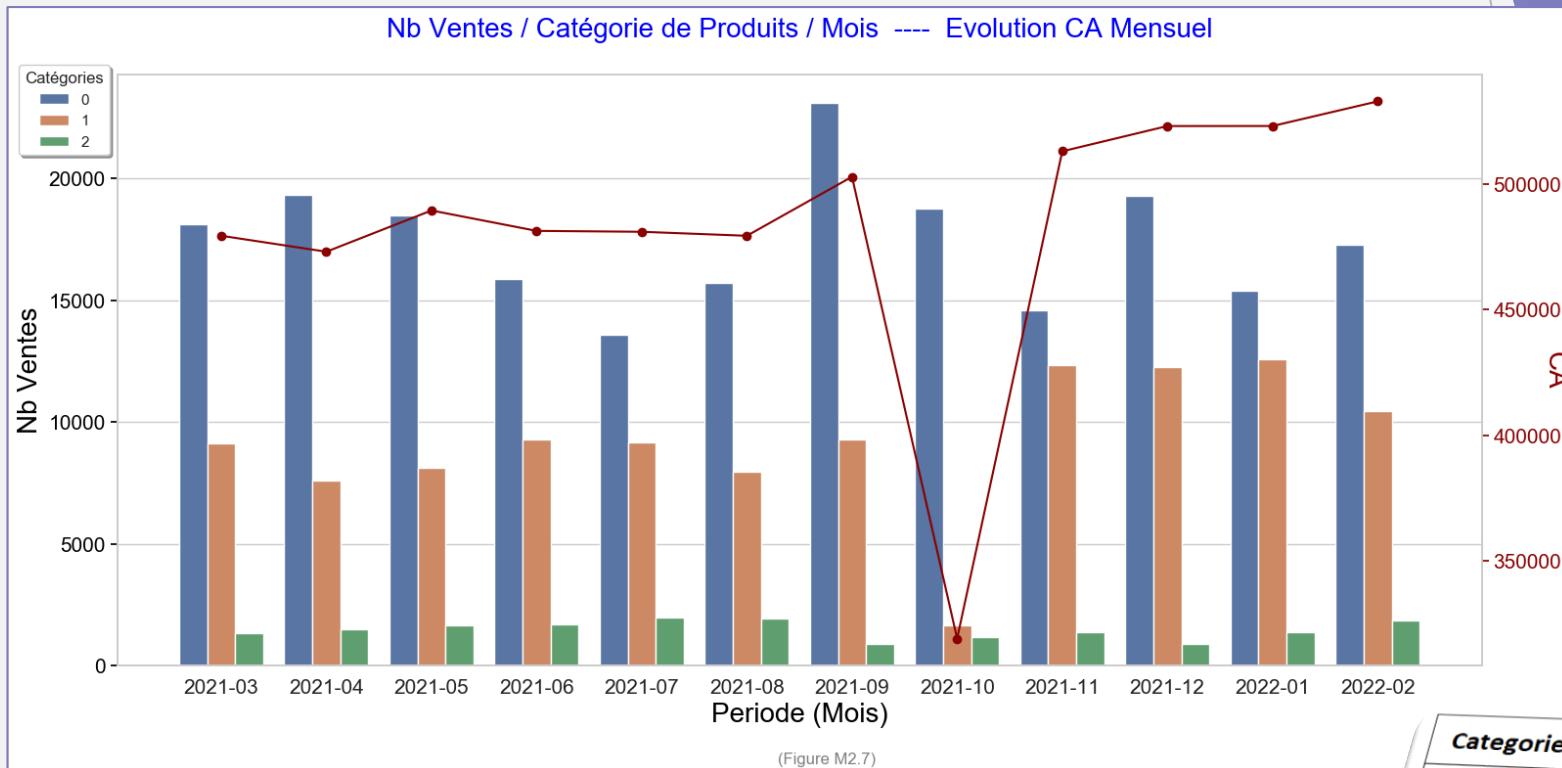
Produit "1_369"

Présent dans les deux tableaux.

Analyses & Graphes

TRANSACTIONS

Evolution du CA Ventes sur 1an & Nombre de Ventes / Mois pour chaque catégorie de produits



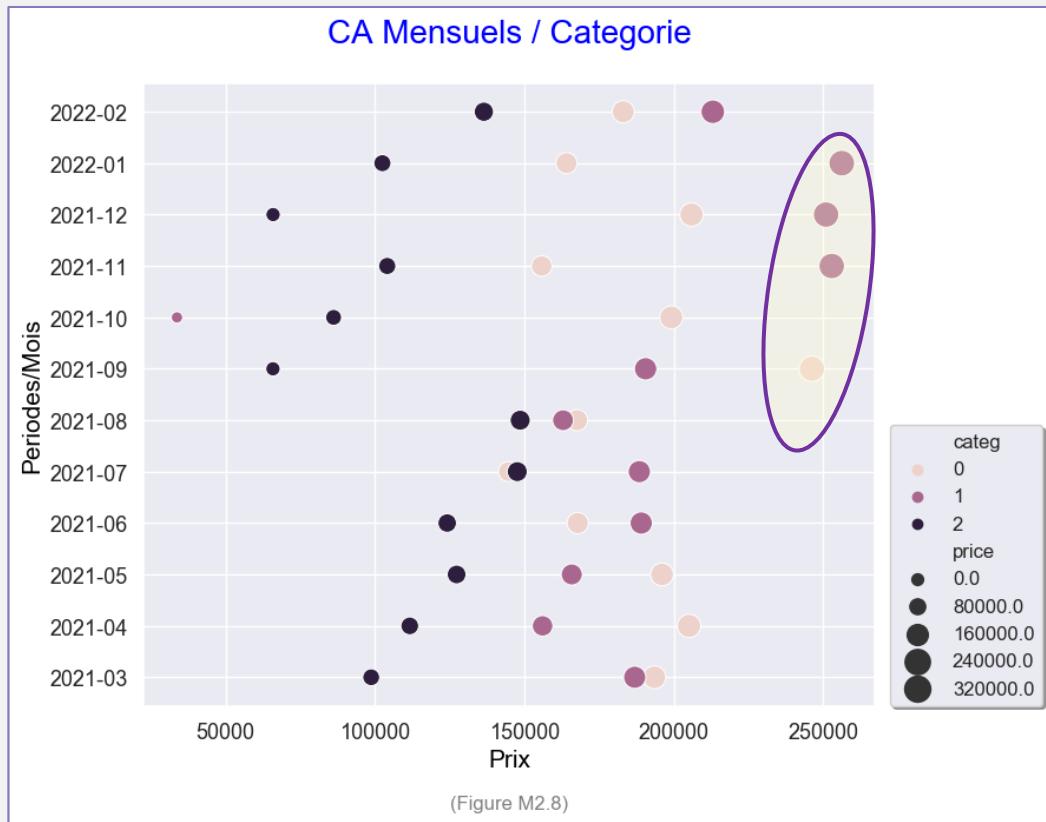
- Forte baisse sur Octobre des ventes de la catégorie « 1 »
- Aucune traces de Ventes dans la table « transactions.csv » entre le 02 et le 27 Octobre
Probablement un **Problème Technique** survenu sur le site. (vu dans la partie 1 nettoyage)
- **Catégorie 0**, « championne » du nombre de ventes → Tarifs des produits les plus bas (0-40€)
Pic Septembre (Rentrée scolaire)

Categorie	Nb Ventes / An
0	209'426
1	109'735
2	17'552

Analyses & Graphes

TRANSACTIONS

Distribution du CA Mensuel des Ventes (€) par Catégorie



- Catégories 0 & 1 à l'honneur: les meilleurs CA mensuels relevés sur l'année

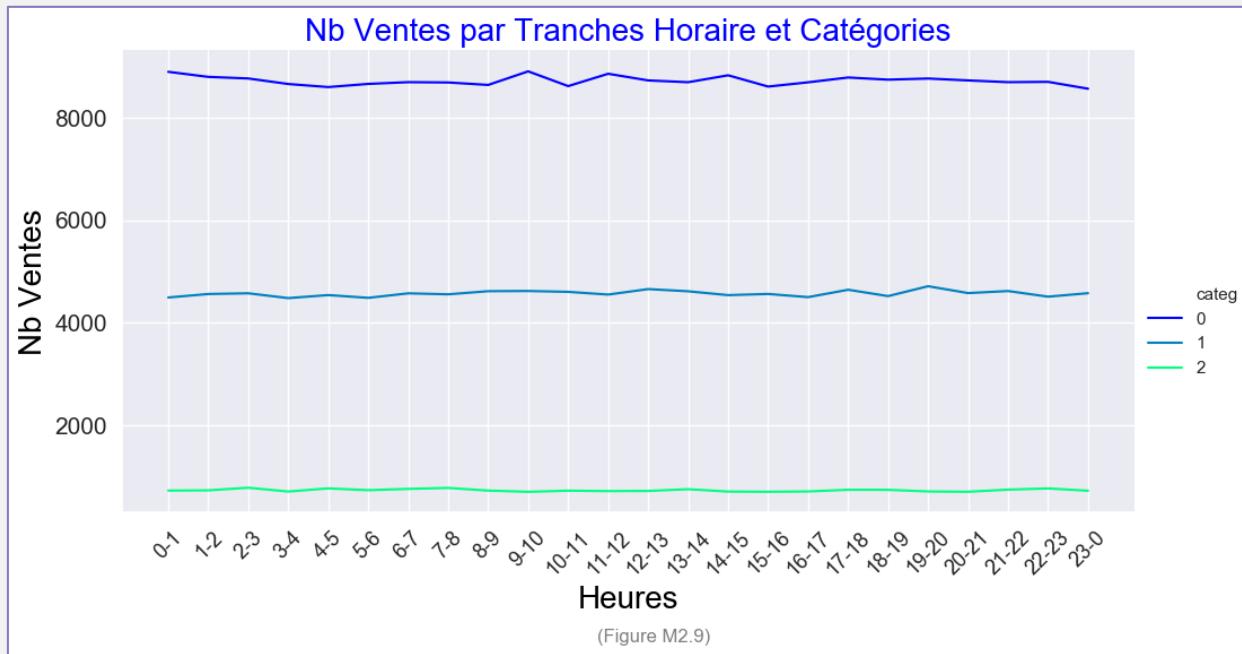
- Septembre : Catégorie « 0 »
"Effet "rentrée scolaire"
- Fin D'année : Catégorie « 1 »
"Effet "Noël et fêtes"

Categorie	CA Annuel (€)
0	2'229'723
1	2'247'384
2	1'319'471

Analyses & Graphes

TRANSACTIONS

Nb Ventes en fonction de l'heure de la journée

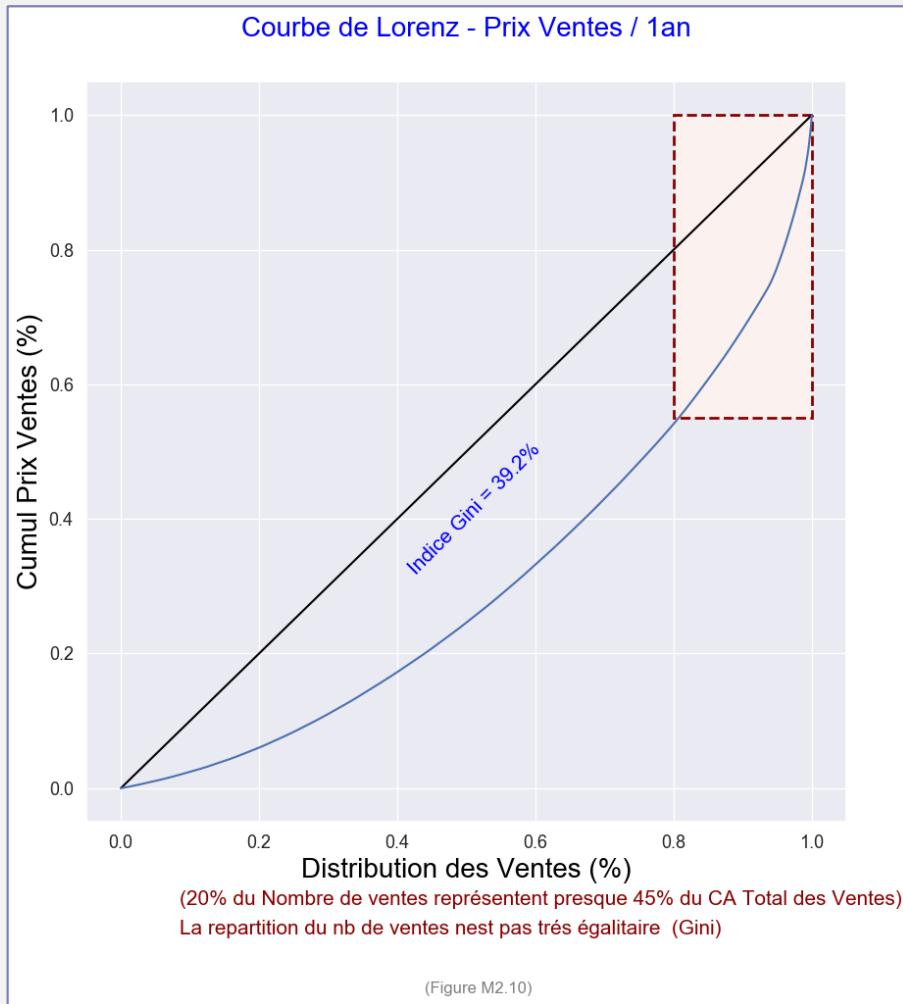


Categ	Nb Ventes (moy / h)
0	23.91
1	12.53
2	2.00

- Grande régularité du nombre de ventes au fil des heures de la journée, quelle que soit la catégorie
- La tranche horaire n'a aucun impact sur le nombre des ventes
- Catégorie 0 domine le nombre des ventes horaire avec une moyenne de presque 24 ventes / h

Analyses & Graphes

Courbe de Lorenz / Indice de Gini Distribution des Ventes Annuelles



Lorenz

Courbe des ventes / ventes cumulées de l'année.

Plus cette courbe est éloignée de la bissectrice, plus les inégalités sont fortes.

Gini

$$G = 2 \times (0,5 - \text{Aire Surface courbe/bissectrice})$$

Entre 0 et 1, l'inégalité est d'autant plus forte que l'indice de Gini est élevé.

*Coefficient
De Gini*
 $C_{gini} = 39,2\%$
Répartition des Ventes
 \Rightarrow peu égalitaire

*20% du Nombre de ventes
représentent
45% du CA Total des Ventes*

Corrélations & Analyses Bivariées

Sexe des Clients / Catégories de Produits

		Question 3.1			CATEGORIE		
		0	1	2			
Quali	SEXE	H	Cpt()	Cpt()	Cpt()		
	CLIENTS	F	Cpt()	Cpt()	Cpt()		

Analyse bivariée sur deux variables « Qualitatives » :

- Appliquer un → Test de χ^2 ([Source Wiki](#))

$$T = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

loi du χ^2 à $(I - 1)(J - 1)$ degrés de liberté.

- Postulat :
- Hypothèse Nulle (H_0) : la catégorie de produits achetée n'est pas liée au sexe des clients
 - Seuil de Tolérance/Risque choisi : $\alpha = 5\%$ (sous l'hypothèse nulle)

Création tableau de contingence

Valeurs Observées (réelles)

Sexe	Cat_0_Reel	Cat_1_Reel	Cat_2_Reel	Tot_Cat_Reel
f	103786	55469	8260	167515
m	105640	54266	9292	169198
total	209426	109735	17552	336713

Valeurs Espérées (attendues) sous H_0

Ratio_Tot	Cat_0_Att	Cat_1_Att	Cat_2_Att
0.4975	104190.0	54593.0	8732.0
0.5025	105236.0	55142.0	8820.0
0.0000	0.0	0.0	0.0

Cumul du nombre de ventes annuelles par catégorie pour chaque sexe.

Total de la catégorie réelle x Ratio calculé du sexe correspondant.

$$(8260 - 8732)^2 / 8732$$

$$T \chi^2 = 81,86$$

Corrélations & Analyses Bivariées

Sexe des Clients / Catégories de Produits

Recherche de notre valeur de χ^2 calculée précédemment dans un abaque avec

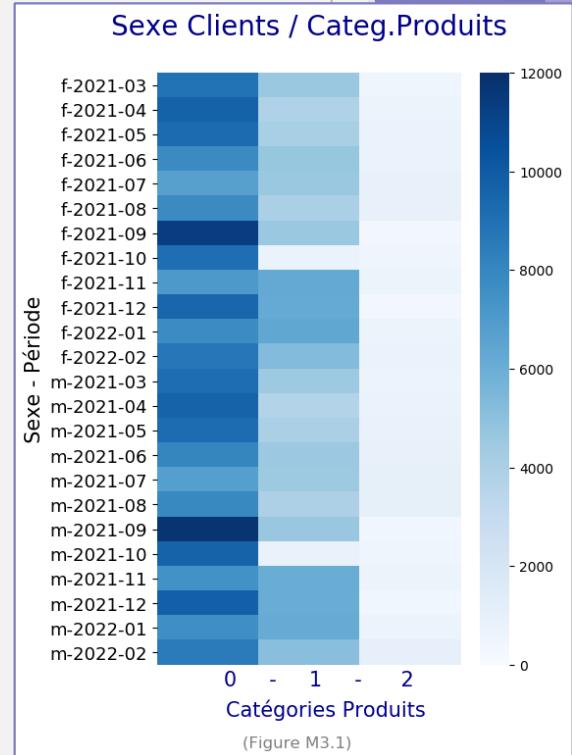
- Degré de liberté = $2 = (\text{Nb lignes} - 1) * (\text{nb colonnes} - 1)$
- Risque $\alpha = 5\%$ (sous hypothèse H_0)

$\chi^2 (81,86) > \text{Valeur Seuil} = 5.99$
L'hypothèse H_0 est donc à rejeter

Table (Source Wiki)

Degrés de liberté	Valeur du χ^2														
1	0.004	0.02	0.06	0.15	0.46	1.07	1.64	2.71	3.84	6.63	10.83				
2	0.10	0.21	0.45	0.71	1.39	2.41	3.22	4.61	5.99	9.21	13.82	81.86		
3	0.35	0.58	1.01	1.42	2.37	3.66	4.64	6.25	7.81	11.34	16.27				
4	0.71	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	13.28	18.47				
5	1.14	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.07	15.09	20.52				
6	1.63	2.20	3.07	3.83	5.35	7.23	8.56	10.64	12.59	16.81	22.46				
7	2.17	2.83	3.82	4.67	6.35	8.38	9.80	12.02	14.07	18.48	24.32				
8	2.73	3.49	4.59	5.53	7.34	9.52	11.03	13.36	15.51	20.09	26.12				
9	3.32	4.17	5.38	6.39	8.34	10.66	12.24	14.68	16.92	21.67	27.88				
10	3.94	4.87	6.18	7.27	9.34	11.78	13.44	15.99	18.31	23.21	29.59				
11	4.57	5.58	6.99	8.15	10.3	12.9	14.6	17.3	19.7	24.7	31.3				
12	5.23	6.30	7.81	9.03	11.3	14.0	15.8	18.5	21.0	26.2	32.9				
13	5.89	7.04	8.63	9.93	12.3	15.1	17.0	19.8	22.4	27.7	34.5				
14	6.57	7.79	9.47	10.8	13.3	16.2	18.2	21.1	23.7	29.1	36.1				
15	7.26	8.55	10.3	11.7	14.3	17.3	19.3	22.3	25.0	30.6	37.7				
$1 - \alpha$	0.05	0.1	0.2	0.3	0.5	0.7	0.8	0.9	0.95	0.99	0.999	1 - p_value		

0.999 < 1 - p_value
p_value < 0.10%



Rappel H_0 : la catégorie de produits achetée n'est pas liée au sexe des clients

La probabilité (p_{value}) pour que l'hypothèse H_0 se réalise est inférieure à 0.10% !!!

La variable "sex des clients" est statistiquement liée à la variable "catégorie de produits" achetés



Corrélations & Analyses Bivariées

Age des Clients / Montant Total Achats

- Analyse bivariée sur deux variables « Quantitatives » :
- Avec *Coefficient de Corrélation linéaire Pearson*

Covariance empirique :

$$s_{X,Y} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

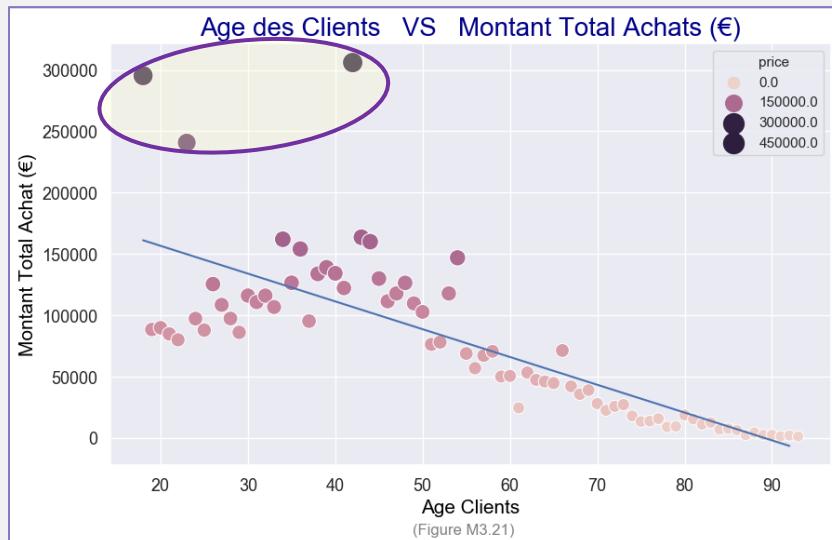
Coefficient de corrélation linéaire (ou « de Pearson ») :

$$r_{X,Y} = \frac{s_{X,Y}}{s_X s_Y}$$

Constitution d'un **Dataframe** pour analyser les variables :

Somme des montants par Age :

- Fonction **Group By** en python avec librairie Pandas



	Question 3.2.1		MONTANT TOTAL ACHATS
	Quanti	Quanti	
AGE CLIENTS	18		Sum()
	19		Sum()
	20		Sum()
	21		Sum()

	92		Sum()
	93		Sum()

Le coefficient de corrélation peut avoir une valeur comprise entre -1 et +1.

Nous utiliserons la fonction « `st.pearsonr` » de la bibliothèque de statistiques `scipy`

```
1 r321 = st.pearsonr(df321["age"],df321["price"])[0]
2 round(r321, 3)
```

r321 = -0.775

L'âge des clients et le montant total des achats, sont donc assez corrélés, surtout à partir de l'âge de 35ans.

Voir expression de la droite de régression linéaire descendante ci-contre.

Le signe négatif du coefficient nous indique le sens de la relation : quand une des deux variables augmente (**montant**), l'autre diminue (**âge**)

Corrélations & Analyses Bivariées

Age des Clients / Fréquence Achats Mensuelle

Analyse bivariée sur deux variables « Quantitatives » :

- Avec *Coefficient de Corrélation linéaire Pearson*

	Question 3.2.2	FREQUENCE_MOY_ACHATS / Mois
		AGE CLIENTS
	18	Avg(Cpt(Mois))
	19	Avg(Cpt(Mois))
	20	Avg(Cpt(Mois))
	21	Avg(Cpt(Mois))
	...	Avg(Cpt(Mois))
	92	Avg(Cpt(Mois))
	93	Avg(Cpt(Mois))

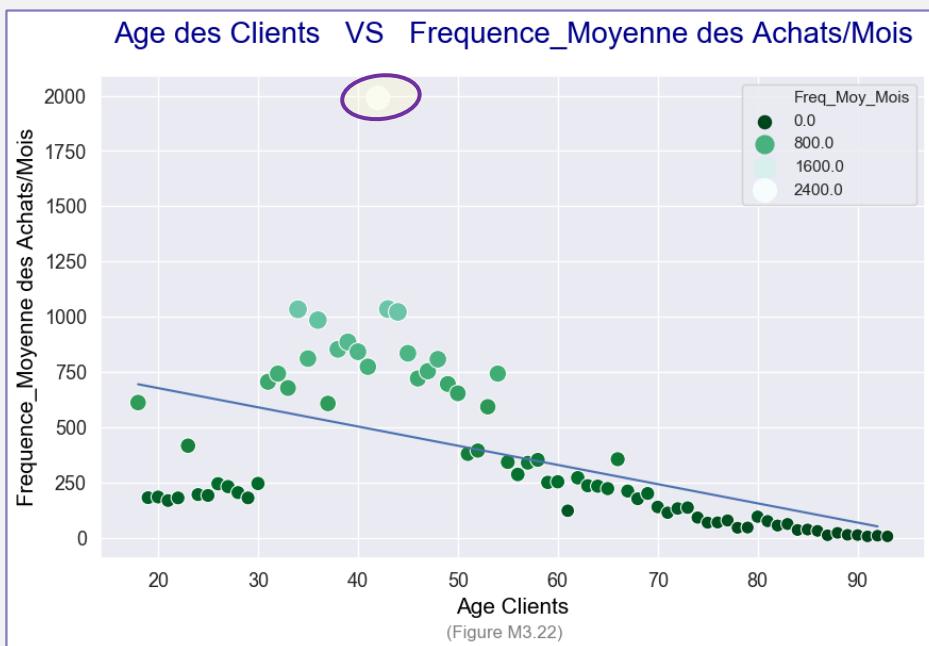
Constitution d'un Dataframe :

- 1°) Nb achats par âge et par Mois
- 2°) Fréquence Moyenne Mensuelle par âge

age	2021-03	2021-04	2021-05	2021-06	2021-07	2021-08	2021-09	2021-10	2021-11	2021-12	2022-01	2022-02	Freq_Moy_Mois
18	596	581	646	645	681	652	521	378	574	653	693	728	612.0
19	207	175	178	164	253	206	150	133	175	138	191	212	182.0
20	136	207	185	258	209	186	180	119	195	156	182	210	185.0

Calculé comme précédemment avec la fonction
« st.pearsonr » de la bibliothèque de statistiques scipy

$$r_{322} = -0.529$$



Les données sont ne sont corrélées qu'à partir de l'âge de 50ans environ.

Un « *outlier* » est à noter autour de l'âge de 42ans (gros client étudié mission 2)

Des corrélation semblent être plus marquées par tranches d'âge :

- 18/30ans
- 30/50ans
- 50+

Corrélations & Analyses Bivariées

Age des Clients / Taille Panier Moyen

- Analyse bivariée sur deux variables « Quantitatives » :
- Avec *Coefficient de Corrélation linéaire Pearson*

Question 3.2.3		TAILLE PANIER MOYEN (Nb Articles)
AGE CLIENTS	18	Avg(Cpt())
	19	Avg(Cpt())
	20	Avg(Cpt())
	21	Avg(Cpt())
	...	Avg(Cpt())
	92	Avg(Cpt())
	93	Avg(Cpt())

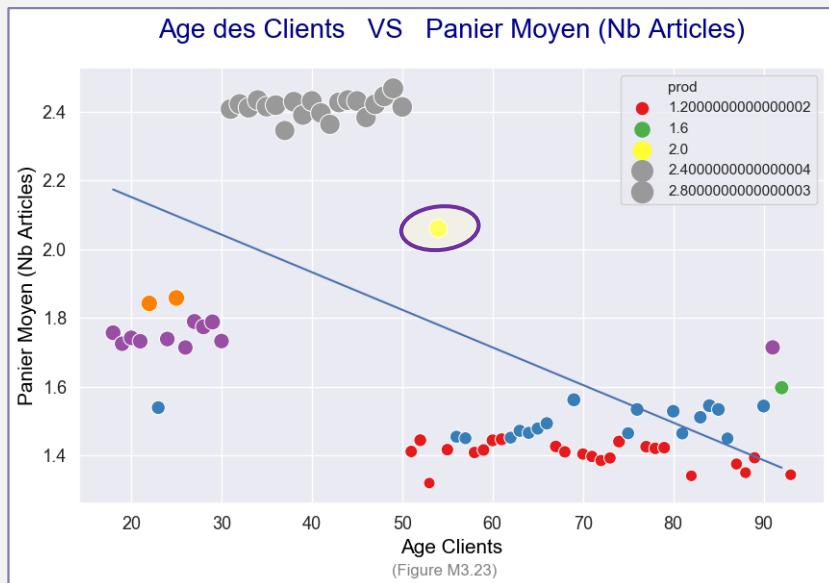
Constitution d'un Dataframe :

- 1°) Nombre d'article par panier <-> nb produits par sessions_id/client
- 2°) Faire la moyenne par "âge" du nombre d'articles

Calculé comme précédemment avec la fonction
 « `st.pearsonr` » de la bibliothèque de statistiques `scipy`

session_id	client	age	prod
s_1	c_329	55	1
s_10	c_2218	52	1
s_100	c_3854	44	2
s_1000	c_1014	33	4
s_10000	c_476	33	3

age	prod
18	1.757054
19	1.724901
20	1.742163
21	1.732310
22	1.842373



$$r_{323} = -0.581$$

Ici on ne peut pas parler de corrélation linéaire.

En revanche la corrélation semble encore plus marquée que précédemment par tranches d'âge :

- 18/30ans
- 30/50ans
- 50+

Un « **outlier** » est à noter autour de l'âge de 53/54ans (gros client étudié mission 2)

Corrélations & Analyses Bivariées

Age des Clients / Catégories de Produits

Ici, analyse bivariée d'une variable « Quantitative » (âge) avec une variable « Quantitative » (catégorie)

Test d'ANOVA (analyse de la variance) avec calcul :

- η² ou R², coefficient de détermination

	Question 3.2.4	CATEGORIE		
		0	1	2
AGE CLIENTS	18	Cpt()	Cpt()	Cpt()
	19	Cpt()	Cpt()	Cpt()
	20	Cpt()	Cpt()	Cpt()
	21	Cpt()	Cpt()	Cpt()
	...	Cpt()	Cpt()	Cpt()
	92	Cpt()	Cpt()	Cpt()
	93	Cpt()	Cpt()	Cpt()

Formule de Décomposition de la Variance			Coefficient de Détermination		
ANOVA			R ²		
(ANAlysis Of VAriance)			$R^2 = \frac{SCE}{SCT}$		
SCT	=SCE	+SCR	Ce coefficient R ² est compris entre [0,1] En effet, 0 ≤ SCE ≤ SCT		
$\sum_{i=1}^n (y_i - \bar{y})^2$	$\sum_{i=1}^n (\hat{y}_i - \bar{y})^2$	$\sum_{i=1}^n (y_i - \hat{y}_i)^2$			

Constitution d'un Dataframe :

Sélection des colonnes :

« âge », « catégorie » et « client »

age	categ	client
45	0	c_4450
41	0	c_5433
37	0	c_857
33	0	c_3679
42	0	c_1609

η² est proche de 0 → Les données ne sont pas corrélées.

L'âge des clients n'a pas d'influence sur la catégorie de produits achetés.

- SCT (Somme des Carrés Totale)
- SCE (Somme des Carrés Expliquée)
- SCR (Somme des Carrés Résiduelle)

Le résultat obtenu est compris entre 0 et 1.
Plus il est proche de 1 plus les données sont corrélées

Calcul (en Python) avec fonction et application sur colonnes « catégorie » et « âge » de notre Dataframe.

```
def eta_squared(x,y):
    moyenne_y = y.mean()
    classes = []
    for classe in x.unique():
        yi_classe = y[x==classe]
        classes.append({'ni': len(yi_classe),
                        'moyenne_classe': yi_classe.mean()})
    SCT = sum([(yj-moyenne_y)**2 for yj in y])
    SCE = sum([c['ni']*(c['moyenne_classe']-moyenne_y)**2 for c in classes])
    return SCE/SCT

# Calcul des eta^2 pour chaque couple
round(eta_squared(df324['categ'],df324['age']), 3)
```

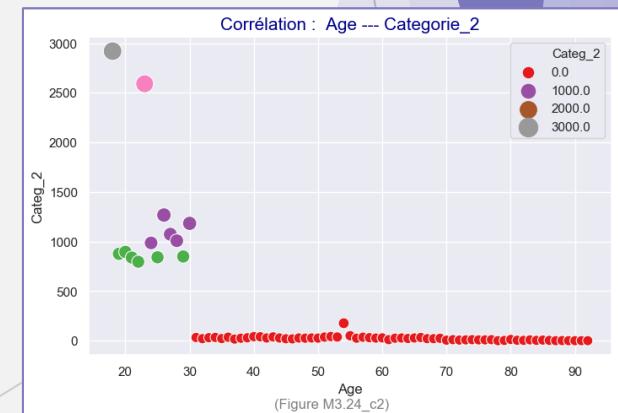
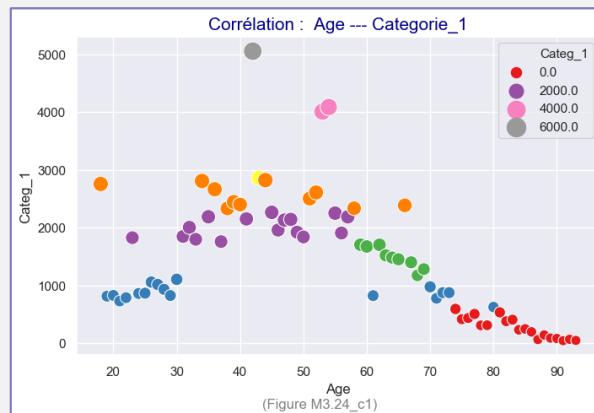
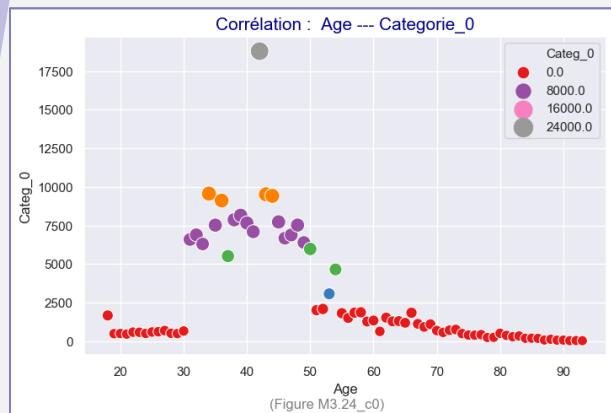
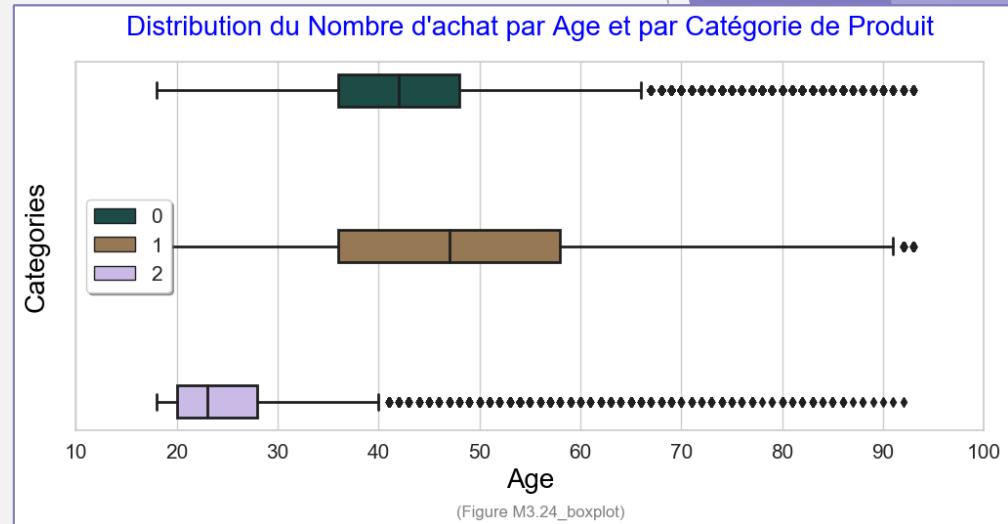
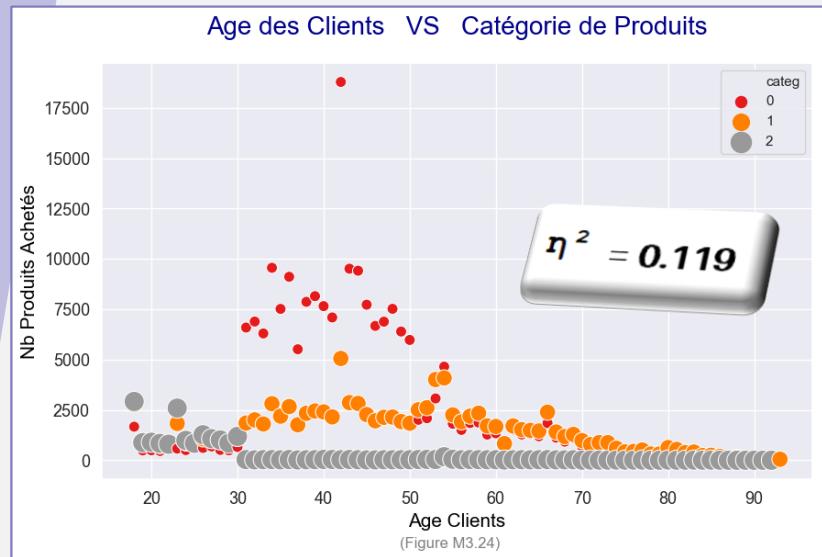
η² = 0.119

Graphes sur la page suivante



Corrélations & Analyses Bivariées

Age des Clients / Catégories de Produits



Analyse des Ventes

Merci pour votre attention !

Je suis à présent disponible pour répondre à vos questions.

Pour plus d'informations, vous pouvez me contacter
en suivant le lien directement ci-dessous

frederic.boissy@gmail.com