Analysez les ventes d'une librairie avec Python

Deuxième partie: Réponses aux analyses demandées

Sommaire

9. Conclusion

1.Importation des bibliothèques et du data frames

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from scipy import stats as sts
from scipy.stats import ttest_1samp
from scipy.stats import f_oneway
from scipy.stats import chi2_contingency
from scipy.stats import chi2
from scipy.stats import pearsonr
import researchpy as rp
import pylab
from scipy.stats import kstest
```

In [2]:
```
# Importation du Dataframe

data_lapage = pd.read_csv('data_lapage.csv')
```

In [3]:
```
# Apercu et dimension du jeu de donnée

print(data_lapage.shape)
data_lapage.head()
```

```
(678512, 17)
```

Out[3]:

	session_id	date_bis	date_annee	date_mois_annee	id_prod	price	categ	client_id
0	s_211425	2022-05-20	2022	2022-05	0_1518	4.18	0.0	c_103
1	s_158752	2022-02-02	2022	2022-02	1_251	15.99	1.0	c_8534
2	s_225667	2022-06-18	2022	2022-06	0_1277	7.99	0.0	c_6714
3	s_52962	2021-06-24	2021	2021-06	2_209	69.99	2.0	c_6941
4	s_325227	2023-01-11	2023	2023-01	0_1509	4.99	0.0	c_4232

Demandes d'Antoine

2. Différents indicateurs et graphiques autour du chiffre d'affaires

In [4]:
```
# Calcul du chiffres d'affaire total

CA = data_lapage['price'].sum()

# Calcul du chiffres d'affaire par an

CA_total = data_lapage.groupby(['date_annee']).sum('price')
CA_total = CA_total[['price']].reset_index()

CA_total.head()
```

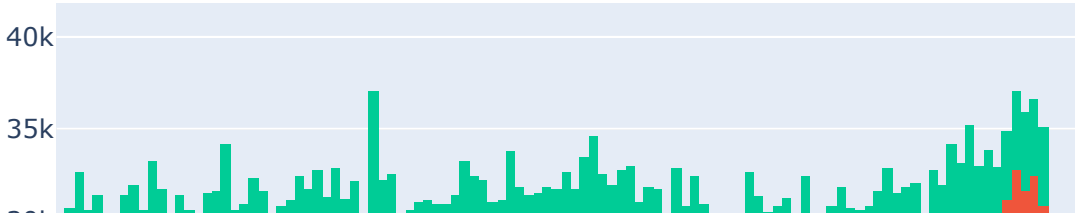
Out[4]:

	date_annee	price
0	2021	4765485.69
1	2022	6100511.00
2	2023	972809.77

In [5]:
```
print("\nLe chiffre d'affaires généré depuis la mise en ligne du site web
      "{:,.2f}".format(CA).replace(',', ' '), '€')
```

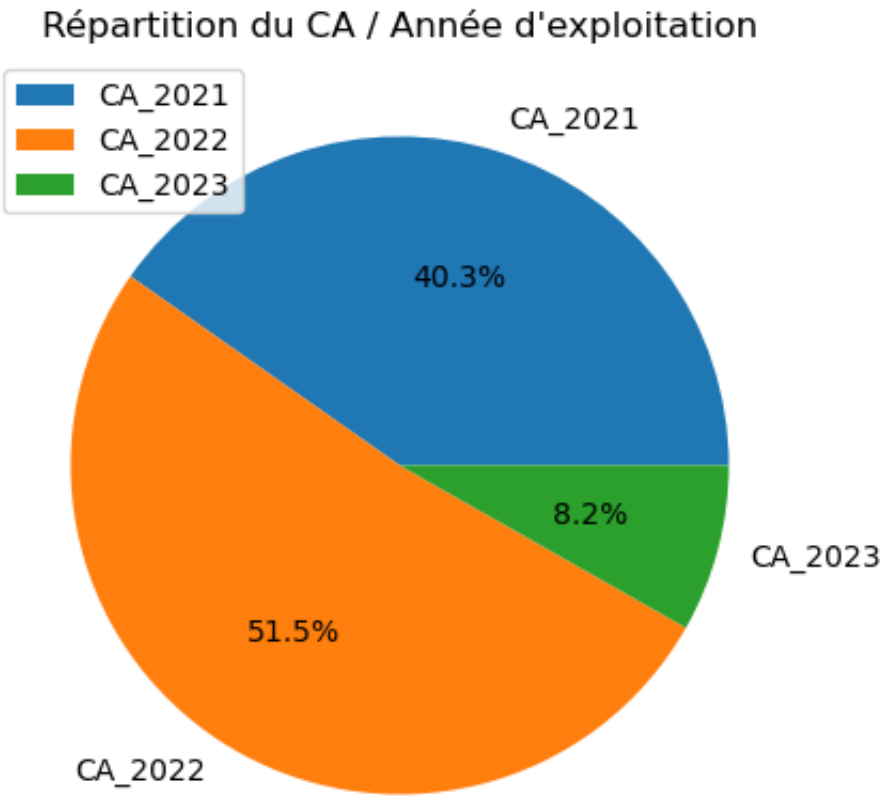
```
Le chiffre d'affaires généré depuis la mise en ligne du site web est 11 8
38 806.46 €
```

In [6]:
```
# Apercu des ventes

df = data_lapage
fig = px.histogram(df, x='date_bis', y='price',
                   height=500, title='', color='categ')
fig.update_layout(title_text='Volume des ventes par catégorie et par mois

fig.show()
```



In [7]:

```
# Répartition du chiffre d'affaires par année

plt.figure(figsize=(20, 5))

plt.pie(CA_total['price'], labels=['CA_2021',
        'CA_2022', 'CA_2023'], autopct='%.1f%%')
plt.title("Répartition du CA / Année d'exploitation")
sns.set_palette('pastel')
plt.legend(loc='upper left')
plt.show()
```



Pour un chiffre d'affaires total de 11 838 806.46 € depuis la mise en place du site , l'année 2022 a été la plus rentable . Cette année, plus de la moitié du chiffre d'affaires globale a été généré, 6 100 511 € soit 51,5 %.

In [8]:
```
# Calcul du chiffres d'affaire par année et par catégorie de produit

CA_categ = data_lapage.groupby(['date_annee', 'categ']).sum('price')

CA_categ = CA_categ[['price']]
```

In [9]:
```
CA_categ['date_annee', 'categ'] = CA_categ.index
CA_categ.reset_index(drop=True).rename(
        columns={'price': 'chiffre_affaires'})
```

Out[9]:

	chiffre_affaires	(date_annee, categ)
0	1881768.92	(2021, 0.0)
1	1775922.05	(2021, 1.0)
2	1107794.72	(2021, 2.0)
3	2191788.81	(2022, 0.0)
4	2482238.62	(2022, 1.0)
5	1426483.57	(2022, 2.0)
6	343427.53	(2023, 0.0)
7	389977.15	(2023, 1.0)
8	239405.09	(2023, 2.0)

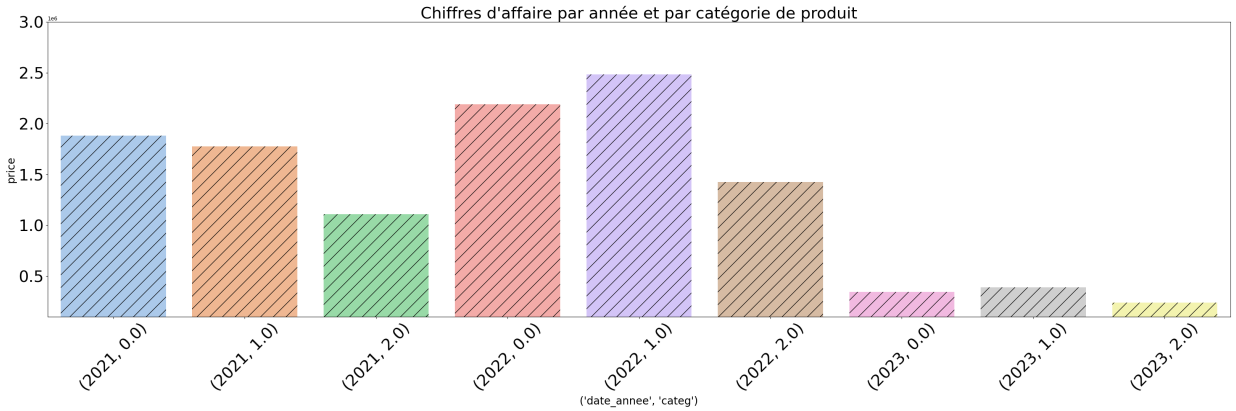
In [10]:

```
# Calcul du chiffres d'affaire par année, par catégorie de produit,
# prix moyen par catégorie et nombre de ventes par catégorie

CA_categ_vente = data_lapage.groupby(['date_annee', 'categ']).agg(
    {'price': ['sum', 'mean'], 'categ': 'count'})

CA_categ_vente
```

Out[10]:

		price		categ
		sum	mean	count
date_annee	**categ**			
2021	**0.0**	1881768.92	10.638555	176882
	1.0	1775922.05	20.505052	86609
	2.0	1107794.72	76.315426	14516
2022	**0.0**	2191788.81	10.636705	206059
	1.0	2482238.62	20.469370	121266
	2.0	1426483.57	76.046677	18758
2023	**0.0**	343427.53	10.634077	32295
	1.0	389977.15	20.506765	19017
	2.0	239405.09	76.979129	3110

In [11]:
```
# Répresentation graphique du CA par année et par catégorie de produit

plt.figure(figsize=(40, 10))
sns.barplot(data=CA_categ, x=('date_annee', 'categ'), y='price', hatch='/
ticks = [1, 10, 100, 1000]
plt.ylim(100000, 3000000)
plt.xticks(rotation=45)
sns.set_palette('dark')
plt.ylabel('price', fontsize=20)
plt.xlabel(('date_annee', 'categ'), fontsize=20);
plt.title("Chiffres d'affaire par année et par catégorie de produit",font
plt.tick_params(axis='both', which='major', labelsize=30)
plt.show()
```



En 2021 et 2022, les catégories 0 et 1 sont ceux qui ont le plus générés de CA, la différence au niveau du chiffre d'affaires n'étaient pas énorme, contrairement au niveau du nombre de vente ou la catégorie 0 est nettement la catégorie la plus vendue.

3. L'évolution dans le temps et mise en place d'une décomposition en moyenne mobile pour évaluer la tendance globale

In [12]:
```
# Calcul du chiffre d'affaires par mois

CA_mois = data_lapage.groupby(['date_mois_annee']).sum('price')

CA_mois = CA_mois[['price']]
CA_mois['date_mois_annee'] = CA_mois.index

CA_mois.reset_index(drop=True)
CA_mois.head()
```

Out[12]:

date_mois_annee	price	date_mois_annee
2021-03	482079.77	2021-03
2021-04	475550.44	2021-04
2021-05	492273.49	2021-05
2021-06	483535.96	2021-06
2021-07	481881.18	2021-07

In [13]:
```
# Calcul du chiffre d'affaires par catégorie de produit

cat0 = data_lapage[data_lapage['categ'] == 0.0]
cat1 = data_lapage[data_lapage['categ'] == 1.0]
cat2 = data_lapage[data_lapage['categ'] == 2.0]

CA_mois0 = cat0.groupby(['date_mois_annee']).sum('price')
CA_mois0 = CA_mois0[['price']]
CA_mois0['date_mois_annee'] = CA_mois0.index
CA_mois0.reset_index(drop=True)


CA_mois1 = cat1.groupby(['date_mois_annee']).sum('price')
CA_mois1 = CA_mois1[['price']]
CA_mois1['date_mois_annee'] = CA_mois1.index
CA_mois1.reset_index(drop=True)


CA_mois2 = cat2.groupby(['date_mois_annee']).sum('price')
CA_mois2 = CA_mois2[['price']]
CA_mois2['date_mois_annee'] = CA_mois2.index
CA_mois2.reset_index(drop=True)

CA_mois2.head()
```

Out[13]:

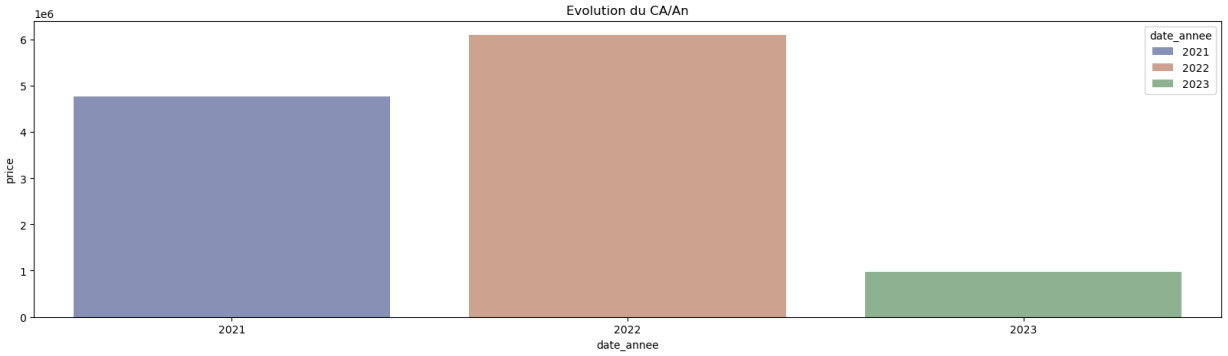
date_mois_annee	price	date_mois_annee
2021-03	101837.27	2021-03
2021-04	114485.15	2021-04
2021-05	130500.41	2021-05
2021-06	126841.08	2021-06
2021-07	148976.06	2021-07

In [14]:
```
# Calcul du chiffre d'affaires par jour

CA_jour = data_lapage.groupby(['date_bis']).sum('price')

CA_jour = CA_jour[['price']]
CA_jour['date_bis'] = CA_jour.index

CA_jour.reset_index(drop=True)
CA_jour.head()
```

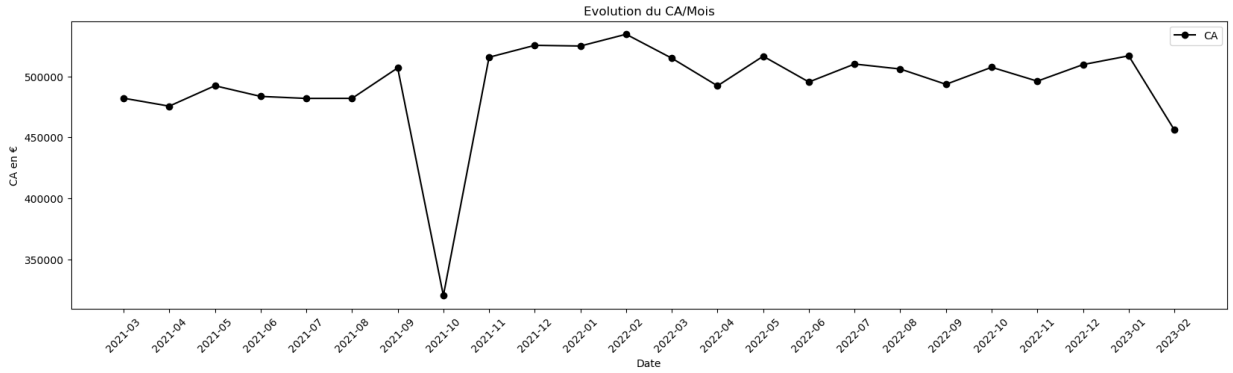
Out[14]:

date_bis	price	date_bis
2021-03-01	16575.21	2021-03-01
2021-03-02	15472.46	2021-03-02
2021-03-03	15165.14	2021-03-03
2021-03-04	15191.18	2021-03-04
2021-03-05	17471.37	2021-03-05

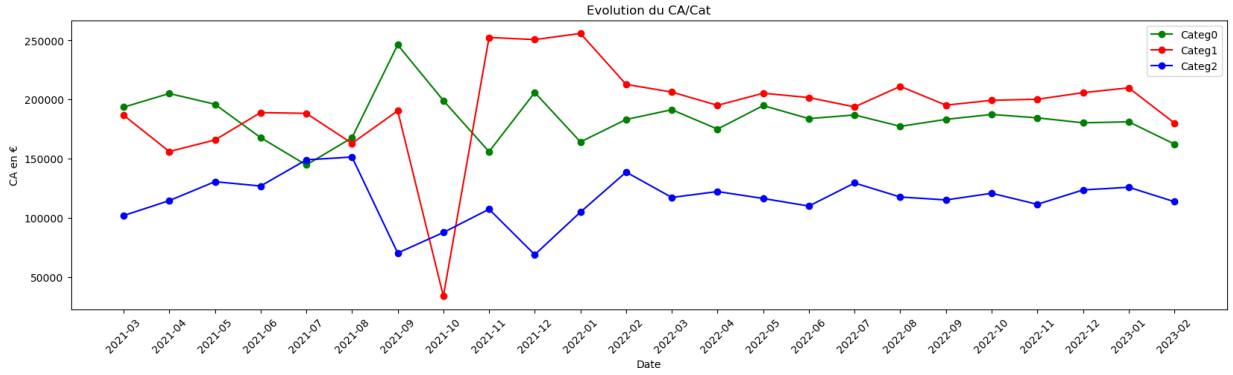
In [15]:
```
# Aperçu de l'évolution du CA/An

plt.figure(figsize=(20, 5))
sns.barplot(data=CA_total, x='date_annee', y='price', hue='date_annee',do

plt.title('Evolution du CA/An');
```



In [16]:
```
# Aperçu de l'évolution du CA/Mois

plt.figure(figsize=(20, 5))
plt.plot(CA_mois['date_mois_annee'], CA_mois['price'],
         marker='o', linestyle='-', color='black', label = 'CA')
plt.xlabel("Date")
plt.xticks(rotation=45)
plt.ylabel("CA en €")
plt.title("Evolution du CA/Mois")
plt.legend();
```



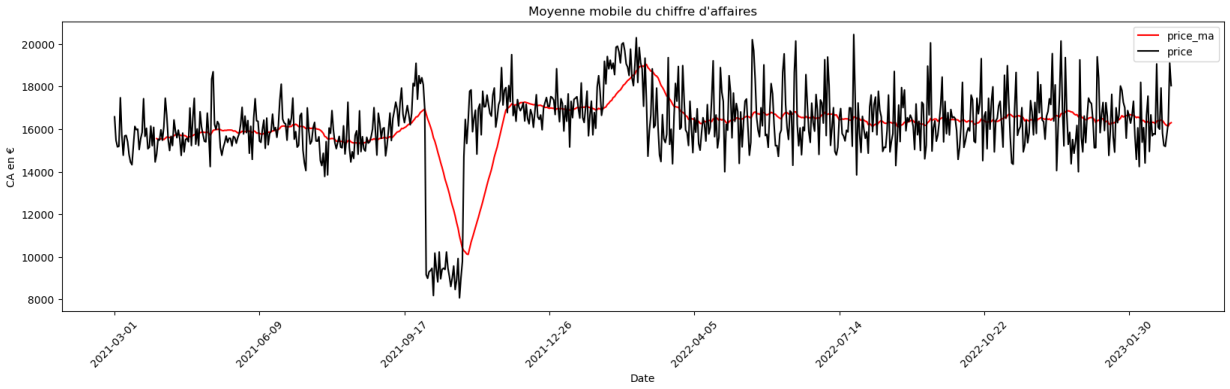
In [17]:
```
# Répresentation graphique du CA par année et par catégorie de produit

plt.figure(figsize=(20, 5))

plt.plot(CA_mois0['date_mois_annee'], CA_mois0['price'],
         marker='o', linestyle='-', color='green', label='Categ0')
plt.plot(CA_mois1['date_mois_annee'], CA_mois1['price'],
         marker='o', linestyle='-', color='red', label='Categ1')
plt.plot(CA_mois2['date_mois_annee'], CA_mois2['price'],
         marker='o', linestyle='-', color='blue', label='Categ2')
plt.xticks(rotation=45)
plt.xlabel("Date")
plt.ylabel("CA en €")
plt.title("Evolution du CA/Cat")
plt.legend();
```



Moyenne mobile

In [18]:
```
# Moyenne mobile

CA_jour['price_ma'] = CA_jour['price'].rolling(window=30).mean()

CA_jour[['price_ma', 'price']].plot(figsize=(20, 5), color=['red', 'black
plt.xlabel("Date")
plt.xticks(rotation=45)
plt.title("Moyenne mobile du chiffre d'affaires")
plt.ylabel("CA en €")
```

Out[18]:
```
Text(0, 0.5, 'CA en €')
```

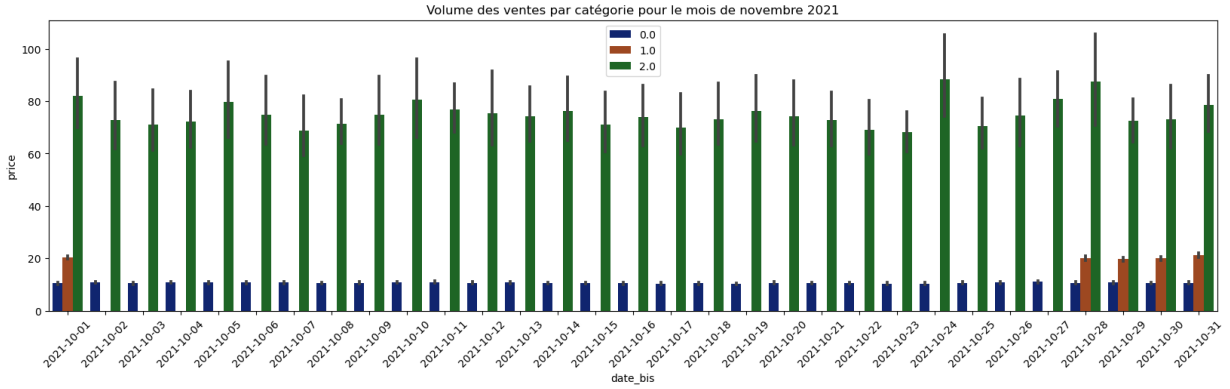


Tendance globale positive et stable du chiffre d'affaires.

Par contre, on observe une grosse chute du chiffre d'affaires pour le mois d'octobre 2021

In [19]:
```
# Zoom sur les ventes pour le mois d'octobre afin de comprendre pourquoi

# Création d'une variable pour extraire les données du mois d'octobre

vente_octobre = data_lapage[data_lapage['date_mois_annee'] == '2021-10']
```

In [20]:
```
plt.figure(figsize=(20, 5))
sns.barplot(data=vente_octobre.sort_values('date_bis', ascending=True), x
plt.xticks(rotation=45)
plt.legend(loc='upper center')
plt.title('Volume des ventes par catégorie pour le mois de novembre 2021'
```



On observe qu'entre le 2 et le 27 octobre 2021, aucune vente pour la cat1 n'a été enregistrée.

N'ayant pas beaucoup d'informations sur cette période, nous pouvons supposer que l'absence de vente peut être due à une rupture de stock pour les articles de cette catégorie.

4. Les tops, les flops, la répartition par catégorie des références

In [21]:
```
# Flop 10 des produits en fonction du nombre de vente

moins_ventes = data_lapage.groupby('id_prod').agg(
    {'price': 'sum', 'session_id': 'count'})
moins_ventes.sort_values('session_id', ascending=True, inplace=True)
moins_ventes = moins_ventes[['session_id']].head(10)


moins_ventes['id_prod'] = moins_ventes.index
moins_ventes.reset_index(drop=True).rename(
    columns={'session_id': 'nombre_vente'})
```

Out[21]:

	nombre_vente	id_prod
0	1	0_549
1	1	0_2201
2	1	2_23
3	1	0_1284
4	1	0_1683
5	1	0_833
6	1	2_98
7	1	0_1633
8	1	0_1601
9	1	2_81

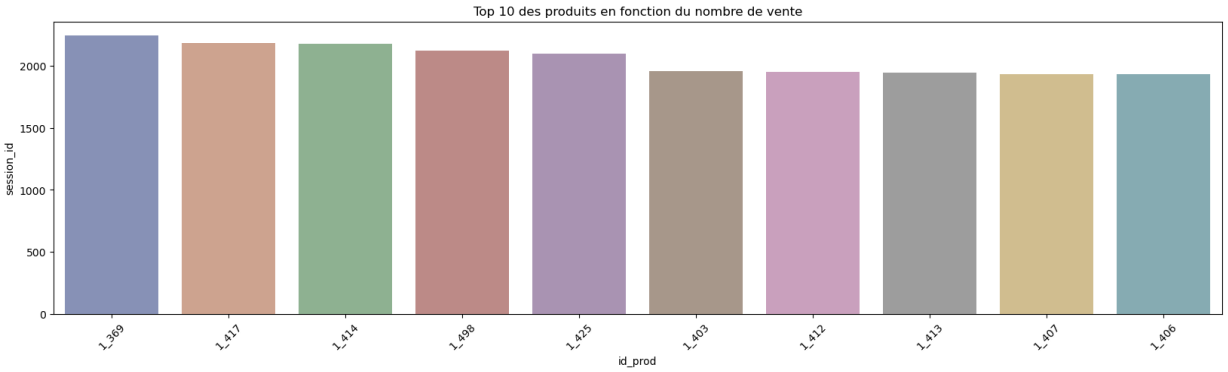
In [22]:
```
# Top 10 des produits en fonction du nombre de vente

plus_ventes = data_lapage.groupby('id_prod').agg(
    {'price': 'sum', 'session_id': 'count'})
plus_ventes.sort_values('session_id', ascending=False, inplace=True)
plus_ventes = plus_ventes[['session_id']].head(10)


plus_ventes['id_prod'] = plus_ventes.index
plus_ventes.reset_index(drop=True).rename(
    columns={'session_id': 'nombre_vente'})
```

Out[22]:

	nombre_vente	id_prod
0	2245	1_369
1	2183	1_417
2	2178	1_414
3	2122	1_498
4	2095	1_425
5	1956	1_403
6	1950	1_412
7	1942	1_413
8	1933	1_407
9	1932	1_406

In [23]:
```
# Répresentation du Top 10 des produits en fonction du nombre de vente

plt.figure(figsize=(20, 5))
sns.barplot(data=plus_ventes, x='id_prod', y='session_id',alpha=0.5)
plt.xticks(rotation=45)
plt.title("Top 10 des produits en fonction du nombre de vente")
```

Out[23]:
```
Text(0.5, 1.0, 'Top 10 des produits en fonction du nombre de vente')
```



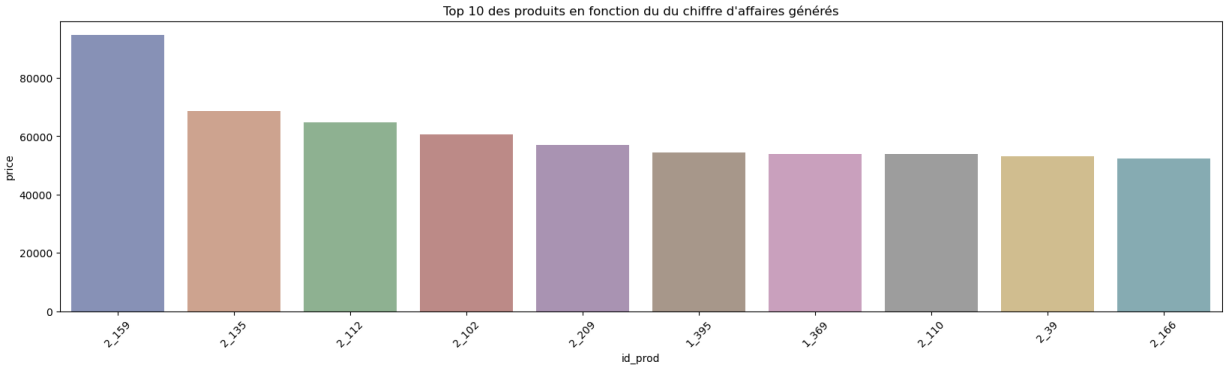
In [24]:
```
# Top des produits en fonction du chiffre d'affaires générés

plus_ventes_ca = data_lapage.groupby('id_prod').agg(
    {'price': 'sum', 'session_id': 'count'})
plus_ventes_ca.sort_values('price', ascending=False, inplace=True)
plus_ventes_ca = plus_ventes_ca[['price']].head(10)


plus_ventes_ca['id_prod']= plus_ventes_ca.index
plus_ventes_ca.reset_index(drop=True).rename(
        columns={'price': 'chiffre_affaires'})
```

Out[24]:

	chiffre_affaires	id_prod
0	94747.51	2_159
1	68576.06	2_135
2	64867.20	2_112
3	60618.50	2_102
4	56971.86	2_209
5	54298.27	1_395
6	53857.55	1_369
7	53846.25	2_110
8	53060.85	2_39
9	52449.12	2_166

In [25]:
```
# Répresentation du Top 10 des produits en fonction du chiffre d'affaires

plt.figure(figsize=(20, 5))
sns.barplot(data=plus_ventes_ca, x='id_prod', y='price',alpha=0.5)
plt.xticks(rotation=45)
plt.title("Top 10 des produits en fonction du du chiffre d'affaires génér
```

Out[25]:
```
Text(0.5, 1.0, "Top 10 des produits en fonction du du chiffre d'affaires
générés")
```

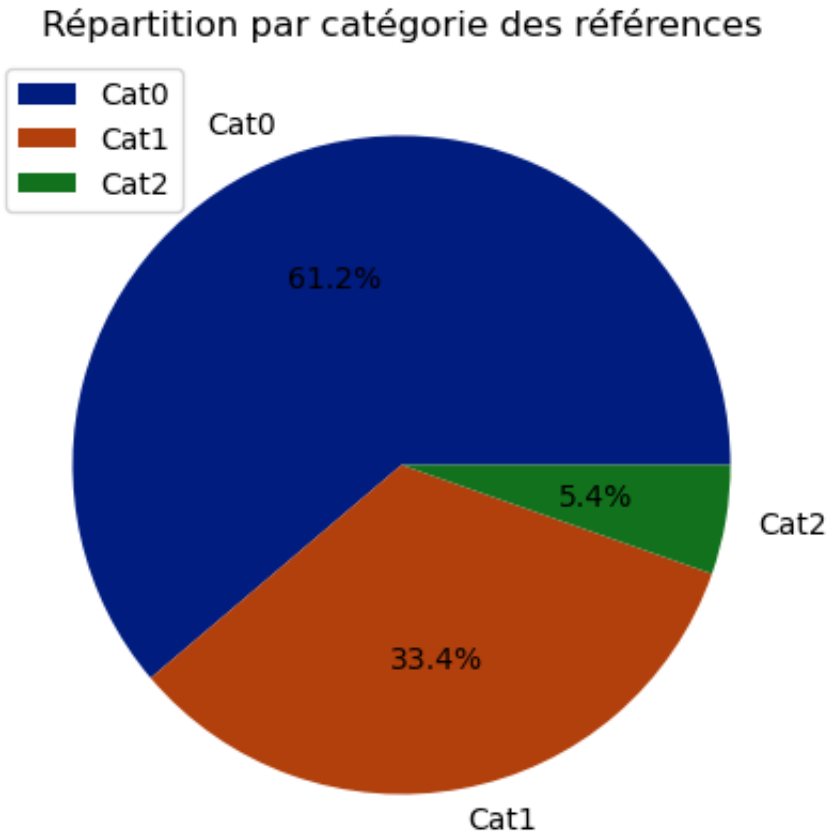


In [26]:
```
# Répartition par catégorie des références

NB_vente = data_lapage.groupby('categ').count()
NB_vente = NB_vente[['session_id']].rename(
    columns={'session_id': 'nombre_vente'})

NB_vente
```

Out[26]:

	nombre_vente
categ	
0.0	415236
1.0	226892
2.0	36384

In [27]:
```
# Répartition par catégorie des références

plt.figure(figsize=(20,5))

plt.pie(NB_vente['nombre_vente'],labels=['Cat0', 'Cat1', 'Cat2'], autopct
plt.title("Répartition par catégorie des références")
sns.set_palette('pastel')
plt.legend(loc='upper left')
plt.show()
```



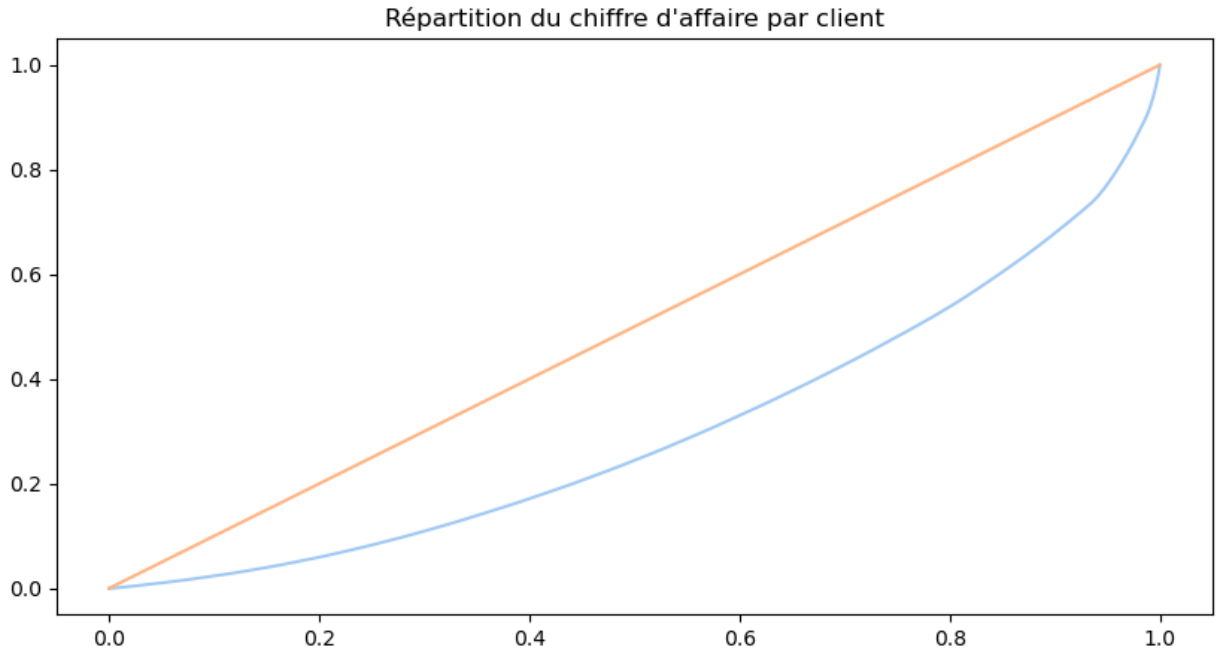
La catégorie 0 est la plus sollicitée avec 61,2% au contraire de la catégorie 2 qui est la moins sollicitée avec 5,37%

5. la répartition du chiffre d'affaires via une courbe de Lorenz

Calcul du coefficient de Gini</u>

Le coefficient de Gini est une mesure statistique permettant de rendre compte de la répartition d'une variable (Exple: salaire, revenus,etc) au sein d'une population. Autrement dit, il mesure le niveau d'inégalité de la répartition d'une variable dans la population. Le coefficient de Gini est un nombre variant de 0 à 1, où 0 signifie l'égalité parfaite et 1, qui ne peut être atteint, signifierait une inégalité parfaite (une seule personne dispose de tous les revenus et une infinité d'autres n'ont aucun revenu).

In [28]:
```
# Courbe de Lorenz

lor = data_lapage[data_lapage['price'] > 0]
dep = lor['price'].values
n = len(dep)
lorenz = np.cumsum(np.sort(dep)) / dep.sum()
lorenz = np.append([0], lorenz)

xaxis = np.linspace(0-1/n, 1+1/n, n+1)
plt.figure(figsize=(10, 5))
plt.plot(xaxis, lorenz, drawstyle='steps-post')
plt.plot(xaxis, xaxis)
plt.title("Répartition du chiffre d'affaire par client")
plt.show()
```



In [29]:
```
# Calcul du coefficient de Gini

# Surface sous la courbe de Lorenz. Le premier segment (lorenz[0]) est à
AUC = (lorenz.sum() - lorenz[-1]/2 - lorenz[0]/2)/n
S = 0.5 - AUC  # surface entre la première bissectrice et le courbe de Lo
gini = 2*S
print("Le coefficient de gini est", gini)
```

```
Le coefficient de gini est 0.3954029193624037
```

Le coefficient de gini étant supérieur à 0, nous pouvons à l'aide de la courbe de Lorenz observer l'inégalité de la répartition du chiffre d'affaires entre les clients.

6. Analyse plus ciblée sur les clients : Comprendre le comportement de nos clients en ligne

In [30]:
```
# Calcul du CA par client

ca_client_vente = data_lapage.groupby(['client_id']).agg(
    {'price': 'sum', 'date_bis': 'count'}).reset_index().rename(
        columns={'date_bis': 'nb_achat','price': 'total_genere'})

ca_client_vente.sort_values('total_genere', ascending=False, inplace=True

# Les 10 clients qui ont générés le plus de chiffre d'affaires
ca_client_vente.head(10)
```

Out[30]:

	client_id	total_genere	nb_achat
677	c_1609	323678.54	25465
4388	c_4958	288600.82	5183
6337	c_6714	153430.54	9175
2724	c_3454	113667.90	6773
2513	c_3263	5276.87	403
634	c_1570	5271.62	369
2108	c_2899	5214.05	105
1268	c_2140	5208.82	402
7006	c_7319	5155.77	371
7791	c_8026	5092.57	377

On observe ici que 4 clients de détachent largement des autres, à savoir les clients 'c_1609', 'c_4958', 'c_6714', 'c_3454', ce sont ceux qui ont le plus générés de CA.

On pourrait donc regrouper les profils clients en deux catégories, à savoir les clients qui font les achats pour eux même et les professionnels

In [31]:
```
# Création de profile client 'Perso' et 'Pro'

data_lapage['type_client'] = 'Perso'
```

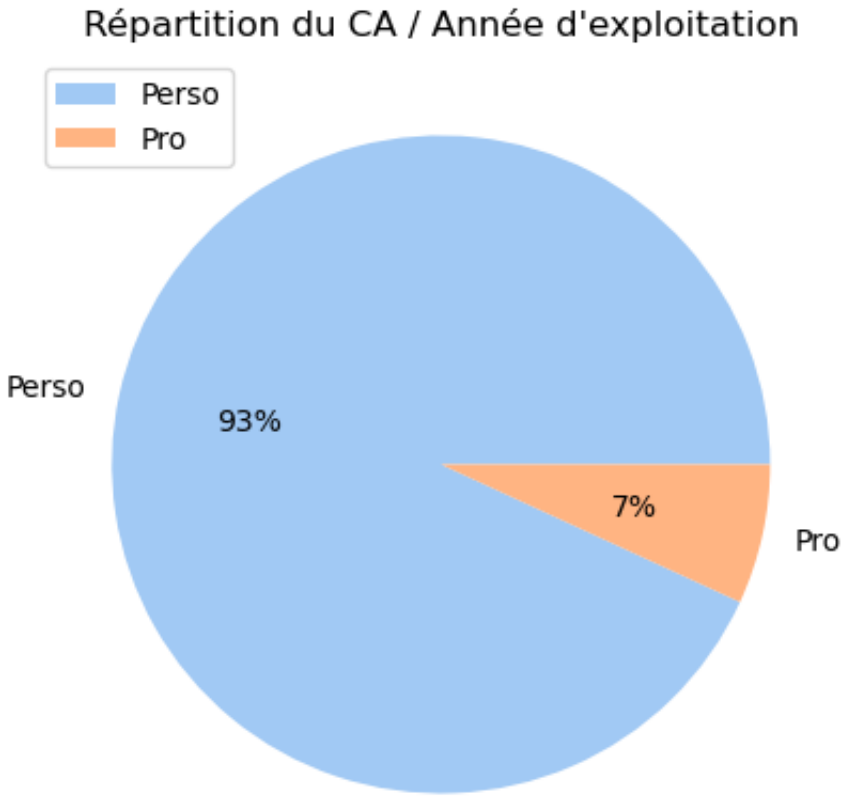
In [32]:
```
liste_pro = ['c_1609', 'c_4958', 'c_6714', 'c_3454']

mask = data_lapage.loc[data_lapage['client_id'].isin(liste_pro)].index

data_lapage.loc[mask, 'type_client'] = 'Pro'
```

In [33]:
```
# Creation de deux data frames, perso et pro pour d'autres analyses

perso = data_lapage[data_lapage['type_client']=='Perso']

pro = data_lapage[data_lapage['type_client']=='Pro']
```

In [34]:
```
print("Les clients pro représentent", round(pro['price'].sum()/data_lapag
      "% de chiffre d'affaires global, soit", "{:,.2f}".format(round(pro[

print("Les clients perso représentent", round(perso['price'].sum()/data_l
)*100), "% de chiffre d'affaires global, soit", "{:,.2f}".format(round(pe
```

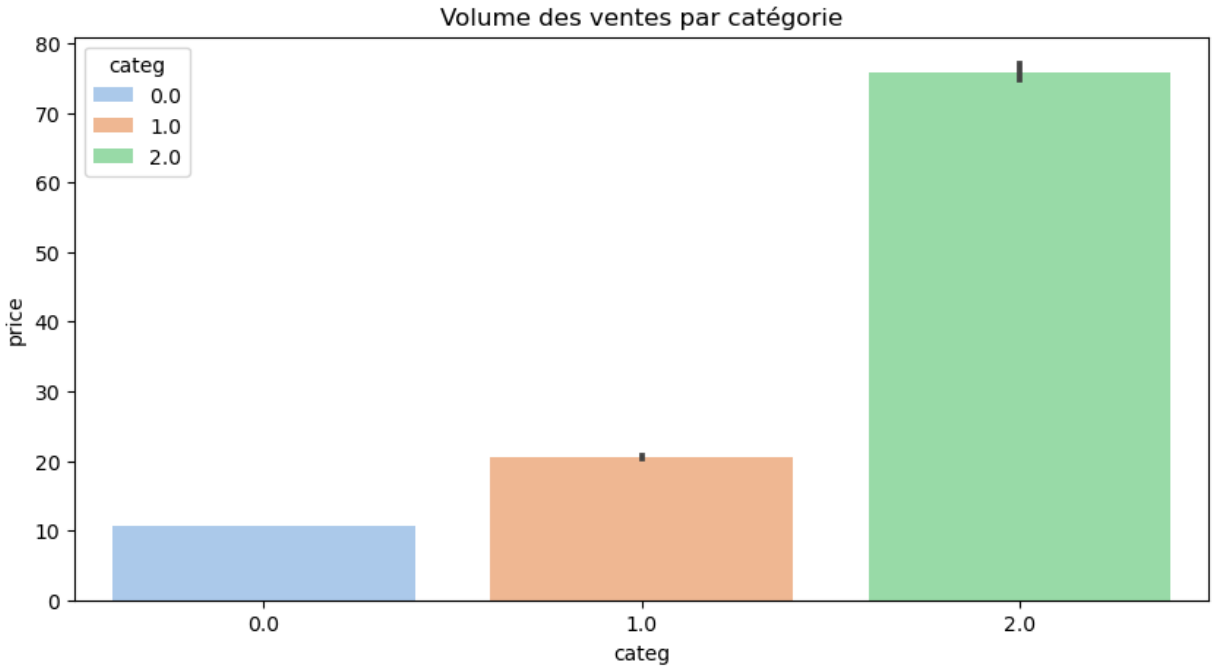
```
Les clients pro représentent 7 % de chiffre d'affaires global, soit 879 3
77.80 €

Les clients perso représentent 93 % de chiffre d'affaires global, soit 10
959 428.66 €
```

In [35]:
```
# Proportion des achats par type de client

plt.figure(figsize=(20, 5))

plt.pie(data_lapage['type_client'].value_counts(normalize=True),labels=['
plt.title("Répartition du CA / Année d'exploitation")
sns.set_palette('pastel')
plt.legend(loc='upper left')
plt.show();
```



Clients Pro

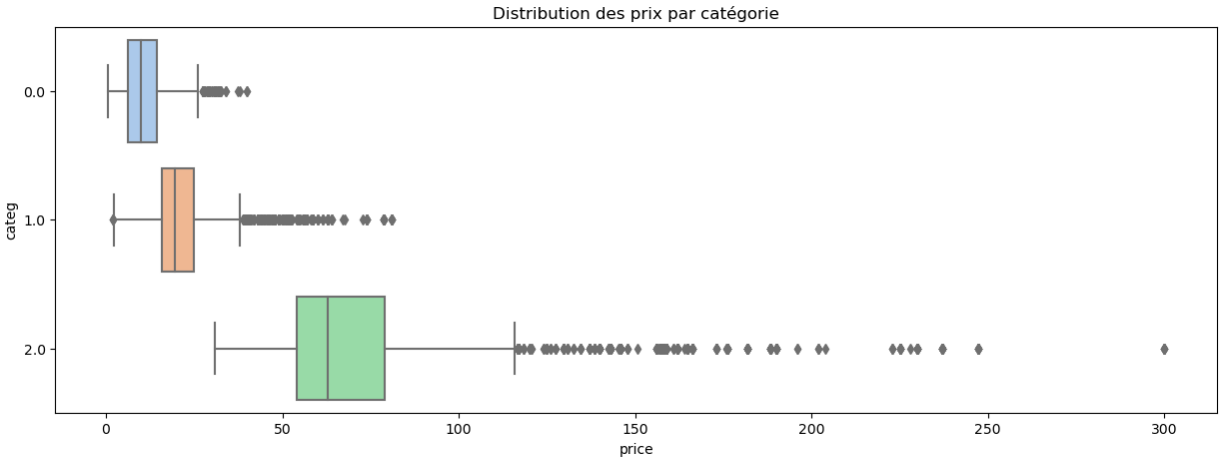
In [36]:
```
# Volume des ventes par catégorie

plt.figure(figsize=(10, 5))
sns.barplot(data=pro, x='categ', y='price',
            hue='categ', dodge=False, alpha=1)
plt.title('Volume des ventes par catégorie')
plt.show()
```



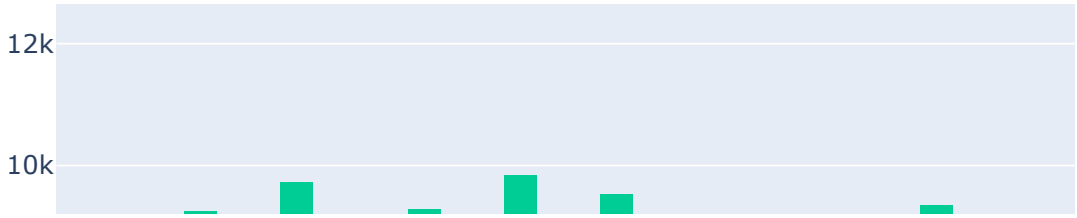
In [37]:

```
# chiffre d'affaires par client et par catégorie

df = pro.sort_values('price', ascending= True)
fig = px.histogram(df, x='categ', y='price',
                   height=500, title='', color='categ')
fig.update_layout(
    title_text="Chiffre d'affaires par catégorie", title_x=0.5)
fig.show()
```

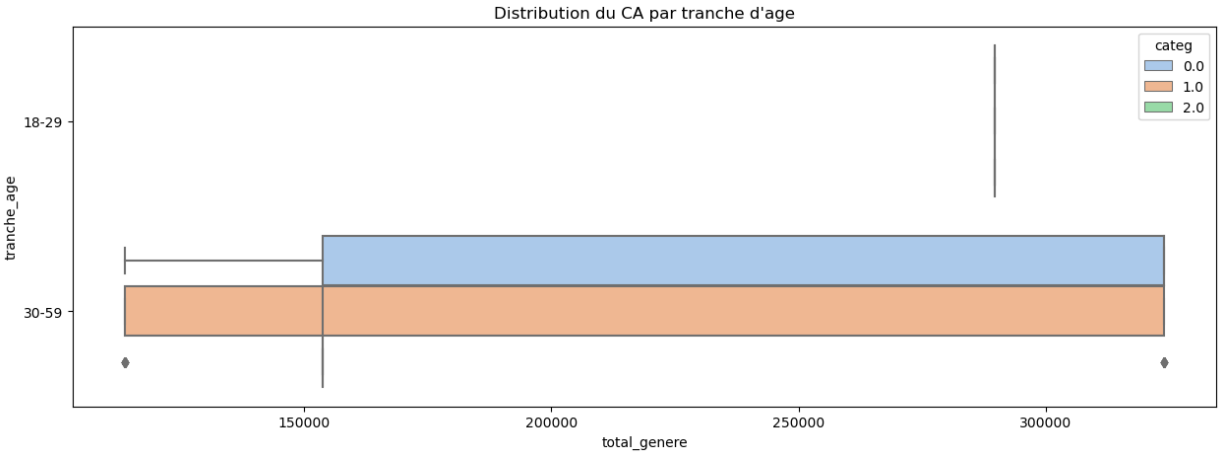
In [38]:
```
# Distribution des prix par catégorie'

plt.figure(figsize=(15, 5))
sns.boxplot(data= pro, x='price', y='categ',orient="h")
plt.title('Distribution des prix par catégorie')
plt.show()
```



In [39]:
```
# Repartition des ventes par catégorie pour les clients pro

df = pro
fig = px.histogram(df, x='date_bis', y='price',
                   height=500, title='', color='categ')
fig.update_layout(
    title_text='Repartition des ventes par catégorie pour les clients pro
fig.show()
```

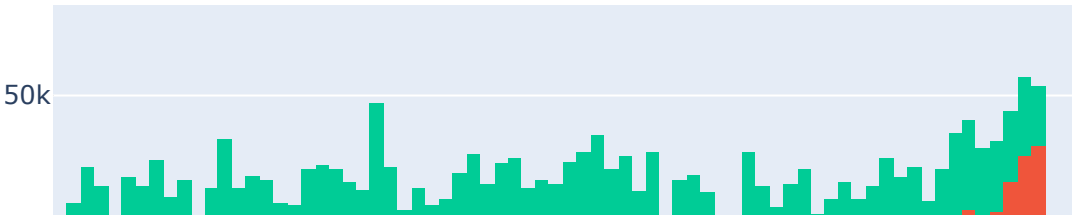


In [40]:
```
# Distribution du CA par tranche d'age

plt.figure(figsize=(15, 5))
sns.boxplot(data= pro.sort_values(by='tranche_age'), x='total_genere', y=
plt.title("Distribution du CA par tranche d'age")
plt.show()
```

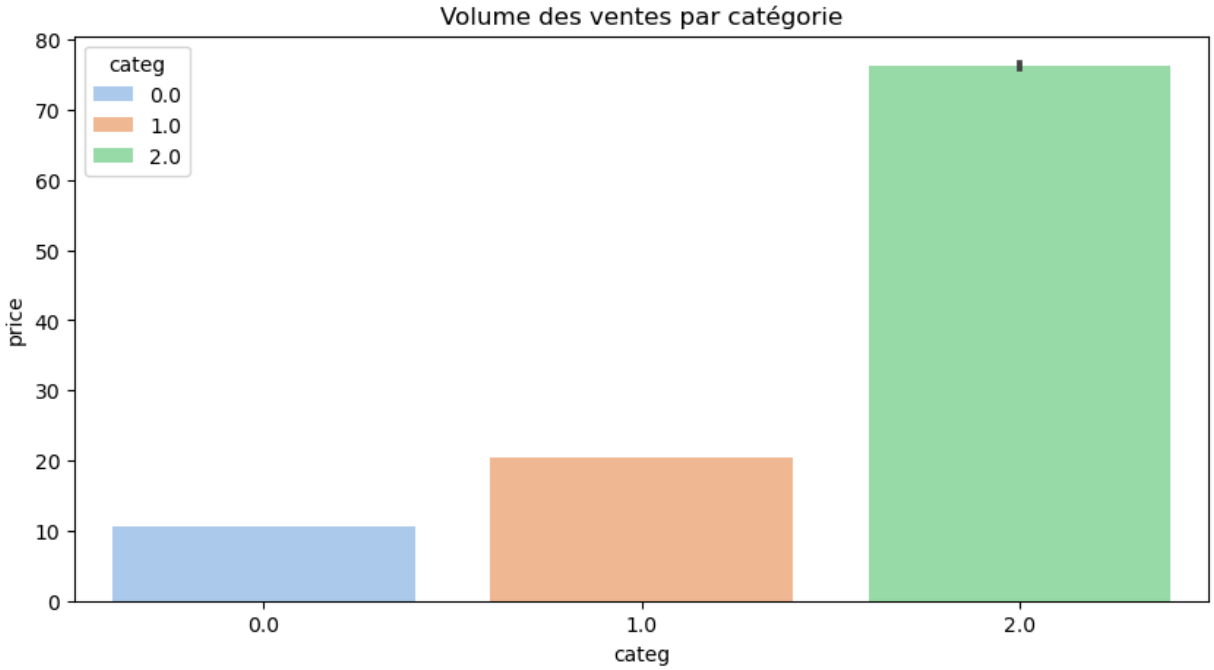


Client Perso

In [41]:
```
# Repartition des ventes par catégorie pour les clients perso

df = perso
fig = px.histogram(df, x='date_bis', y='price',
                   height=500, title='', color='categ')
fig.update_layout(
    title_text='Repartition des ventes par catégorie pour les clients per
fig.show()
```

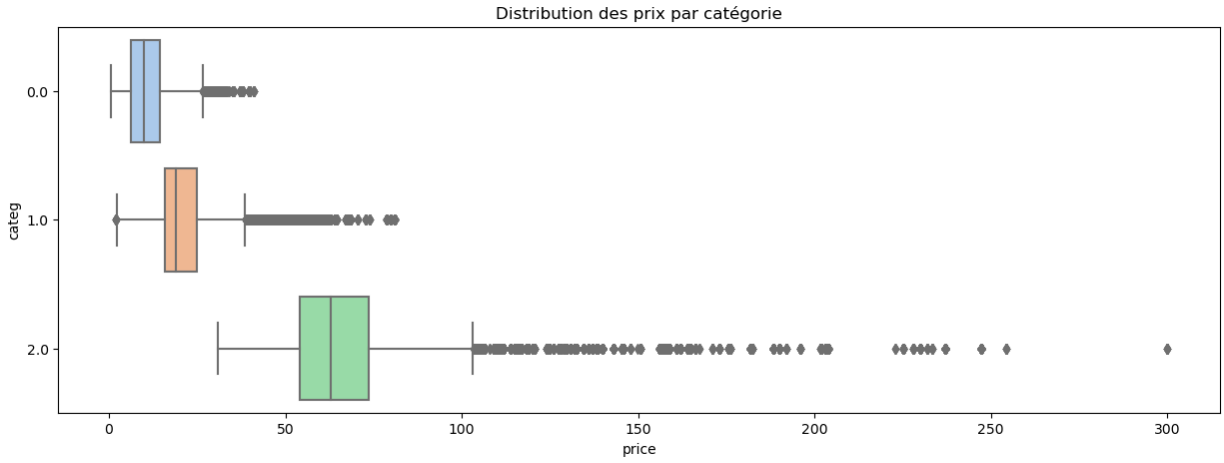


In [42]:
```
# Volume des ventes par catégorie

plt.figure(figsize=(10, 5))
sns.barplot(data=perso, x='categ', y='price',
            hue='categ', dodge=False, alpha=1)
plt.title('Volume des ventes par catégorie')
plt.show()
```

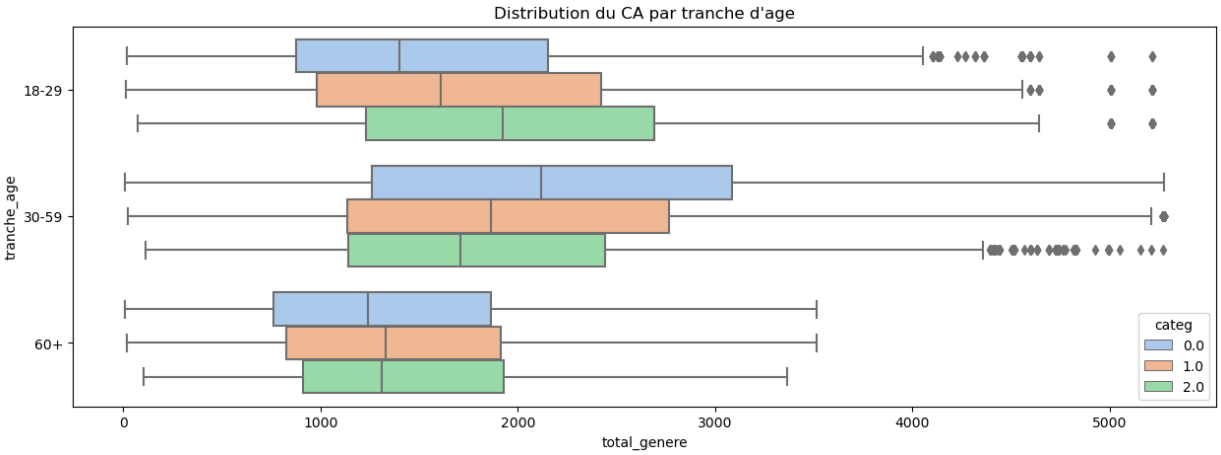


In [43]:
```
# Chiffre d'affaires par catégorie

df = perso
fig = px.histogram(df, x='categ', y='price',
                   height=500, title='', color='categ')
fig.update_layout(
    title_text="Chiffre d'affaires par catégorie", title_x=0.5)
fig.show()
```

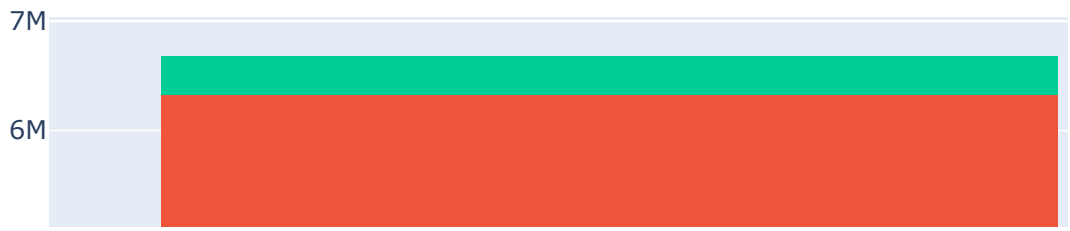
In [44]:
```
# Distribution des prix par catégorie

plt.figure(figsize=(15, 5))
sns.boxplot(data=perso, x=perso['price'], y=perso['categ'], orient="h")
plt.title('Distribution des prix par catégorie')
plt.show();
```



In [45]:
```
# Distribution du CA par tranche d'age

plt.figure(figsize=(15, 5))
sns.boxplot(data=perso.sort_values(by='tranche_age'),
            x='total_genere', y='tranche_age', hue='categ')
plt.title("Distribution du CA par tranche d'age")
plt.show()
```



In [46]:
```
# Chiffre d'affaires par tranche d'age

df = perso
fig = px.histogram(df, x='tranche_age', y='price',
                   height=500, title='', color='categ')
fig.update_layout(
    title_text="Chiffre d'affaires par tranche d'age", title_x=0.5)
fig.show()
```



In [47]:
```
# Calcul du CA par tranche_age

ca_tranche_age = data_lapage.groupby(['tranche_age','categ']).agg(
    {'price': 'sum', 'date_bis': 'count', 'date_bis': 'count'}).reset_ind
        columns={'date_bis': 'nb_achat','price': 'total_genere'})

ca_tranche_age.sort_values('tranche_age', ascending=True, inplace=True)


ca_tranche_age.head(10)
```

Out[47]:

	tranche_age	categ	total_genere	nb_achat
0	18-29	0.0	164373.54	15434
1	18-29	1.0	569297.29	27730
2	18-29	2.0	2324389.94	30741
3	30-59	0.0	3832662.43	360471
4	30-59	1.0	3056578.52	149259
5	30-59	2.0	382621.92	4804
6	60+	0.0	419949.29	39331
7	60+	1.0	1022262.01	49903
8	60+	2.0	66671.52	839

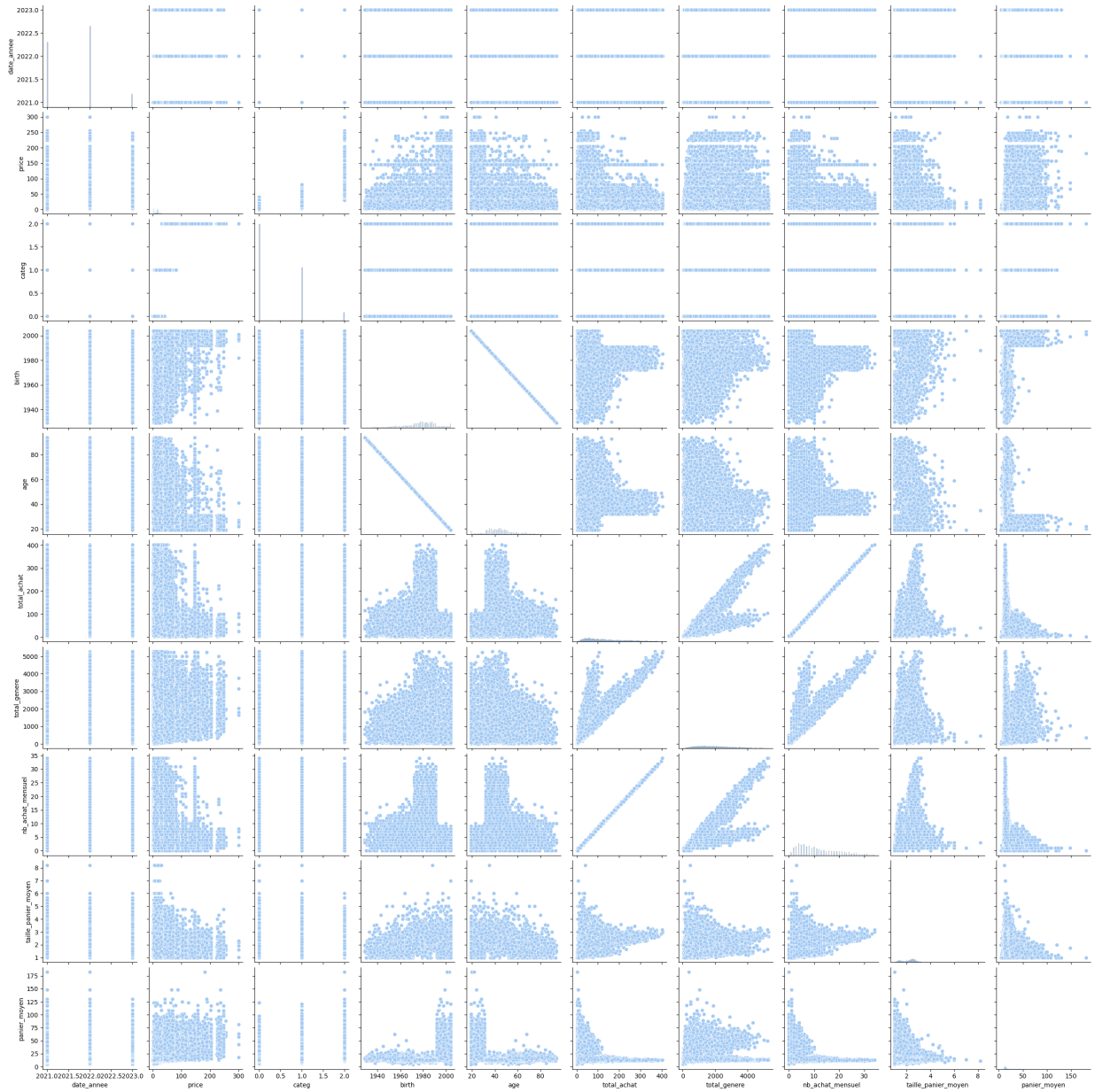
Chez les 18-29 ans, c'est la catégorie 2 qui a été la plus vendue et qui à générée le plus de chiffre d'affaire, au niveau des 30-59 ans, c'est la catégorie 0 qui a été la plus vendue et la plus génératrice de chiffre d'affaires, et afin chez les plus de 60 ans, c'est la catégorie 1 qui a été la plus vendue et la plus rentable.

In [48]:
```
# Matrice de corrélation
perso.corr()
```

Out[48]:

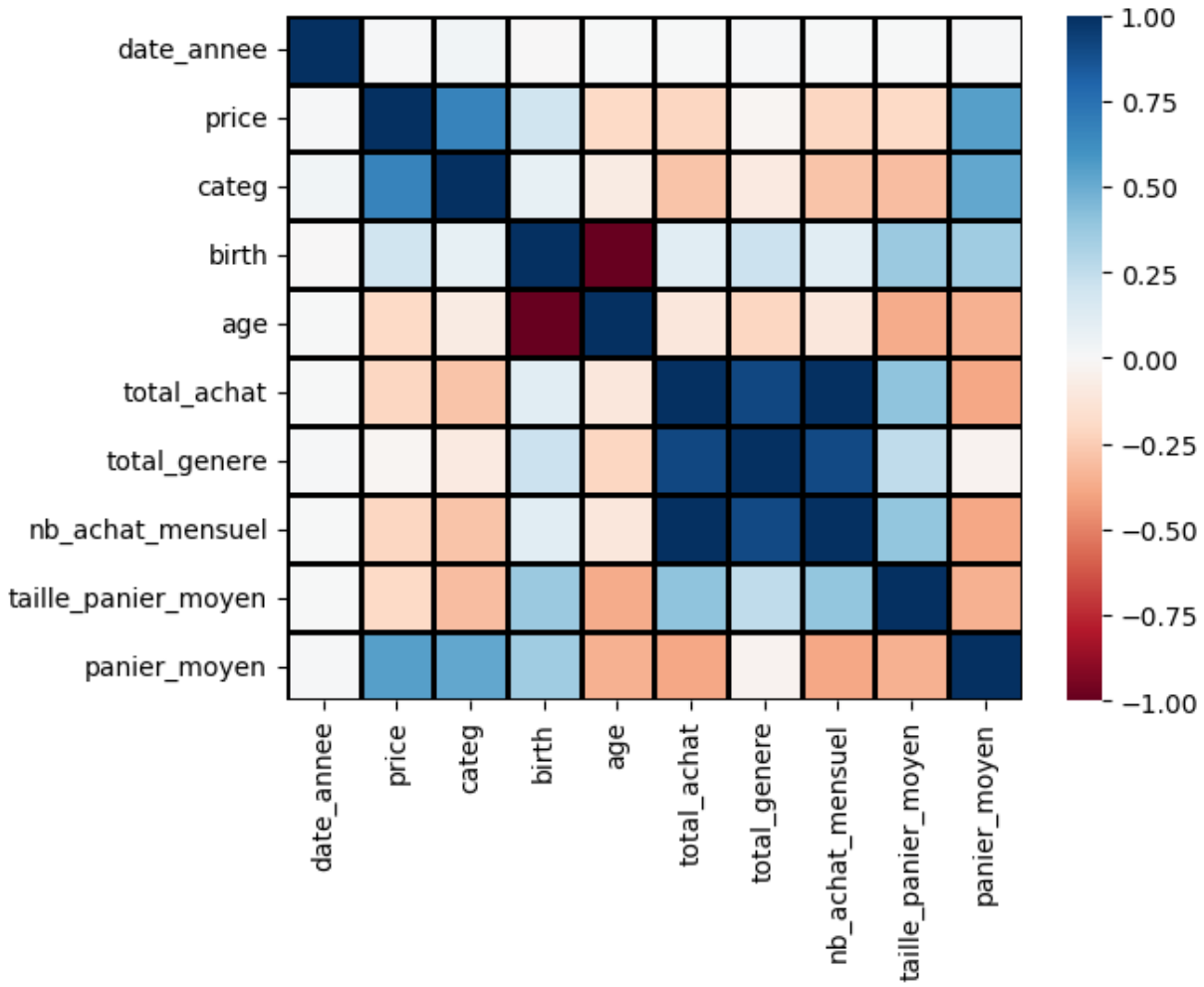
	date_annee	price	categ	birth	age	total_acha
date_annee	1.000000	0.014265	0.033412	-0.002574	0.002574	0.00370
price	0.014265	1.000000	0.669194	0.197524	-0.197524	-0.21398
categ	0.033412	0.669194	1.000000	0.084176	-0.084176	-0.28456
birth	-0.002574	0.197524	0.084176	1.000000	-1.000000	0.11337
age	0.002574	-0.197524	-0.084176	-1.000000	1.000000	-0.11337
total_achat	0.003705	-0.213989	-0.284566	0.113375	-0.113375	1.00000
total_genere	0.012326	-0.018855	-0.089935	0.218656	-0.218656	0.90647
nb_achat_mensuel	0.003685	-0.213812	-0.284308	0.112426	-0.112426	0.99925
taille_panier_moyen	0.005371	-0.195656	-0.304882	0.373780	-0.373780	0.39920
panier_moyen	0.008862	0.550278	0.519720	0.358869	-0.358869	-0.38892

In [49]:
```
sns.pairplot(perso);
```



In [50]:

```
# Heatmap

sns.heatmap(perso.corr(), cmap='RdBu',linewidths=1,linecolor='black');
```



Demandes Julie

7. Lien entre le genre d'un client et les catégories des livres achetés

Nous allons chercher confirmer à travers le test de Chi2 à travers les hypothèses suivantes:

H0 : Il n'y a pas de lien entre le genre et la catégorie de livre achetée.

H1 : On rejette H0: Il n'y a un lien entre le genre et la catégorie de livre achetée.

Seuil α = 0.05

Si pvalue > α, on accepte H0

Si pvalue < α, on rejette H0

In [51]:
```
pip install researchpy
```

```
Requirement already satisfied: researchpy in /Users/yacou/opt/anaconda3/l
ib/python3.9/site-packages (0.3.5)
Requirement already satisfied: scipy in /Users/yacou/opt/anaconda3/lib/py
thon3.9/site-packages (from researchpy) (1.9.1)
Requirement already satisfied: patsy in /Users/yacou/opt/anaconda3/lib/py
thon3.9/site-packages (from researchpy) (0.5.2)
Requirement already satisfied: statsmodels in /Users/yacou/opt/anaconda3/
lib/python3.9/site-packages (from researchpy) (0.13.2)
Requirement already satisfied: numpy in /Users/yacou/opt/anaconda3/lib/py
thon3.9/site-packages (from researchpy) (1.21.5)
Requirement already satisfied: pandas in /Users/yacou/opt/anaconda3/lib/p
ython3.9/site-packages (from researchpy) (1.4.4)
Requirement already satisfied: python-dateutil>=2.8.1 in /Users/yacou/opt
/anaconda3/lib/python3.9/site-packages (from pandas->researchpy) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /Users/yacou/opt/anaconda3
/lib/python3.9/site-packages (from pandas->researchpy) (2022.1)
Requirement already satisfied: six in /Users/yacou/opt/anaconda3/lib/pyth
on3.9/site-packages (from patsy->researchpy) (1.16.0)
Requirement already satisfied: packaging>=21.3 in /Users/yacou/opt/anacon
da3/lib/python3.9/site-packages (from statsmodels->researchpy) (21.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /Users/yacou/o
pt/anaconda3/lib/python3.9/site-packages (from packaging>=21.3->statsmode
ls->researchpy) (3.0.9)
Note: you may need to restart the kernel to use updated packages.
```

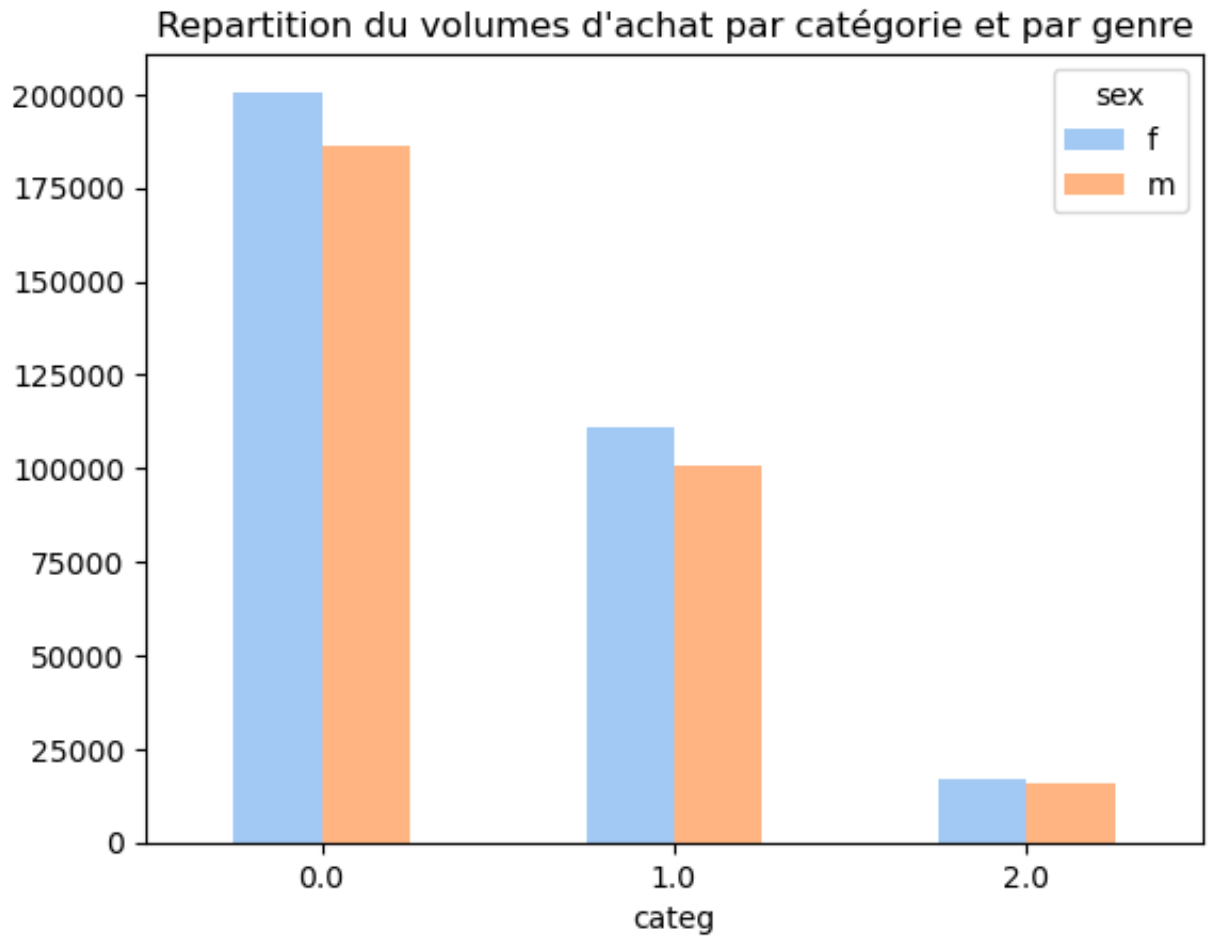
In [52]:
```
# Création d'une table de contingence

table_contingence = pd.crosstab(
    perso['categ'], perso['sex'])

table_contingence
```

Out[52]:

sex	f	m
categ		
0.0	200659	186412
1.0	111199	100886
2.0	16953	15807

In [53]:
```
# Aperçu de la table de contingence

table_contingence.plot.bar()
plt.title("Repartition du volumes d'achat par catégorie et par genre")
plt.xticks(rotation=0)
```

Out[53]:
```
(array([0, 1, 2]), [Text(0, 0, '0.0'), Text(1, 0, '1.0'), Text(2, 0, '2.0
')])
```



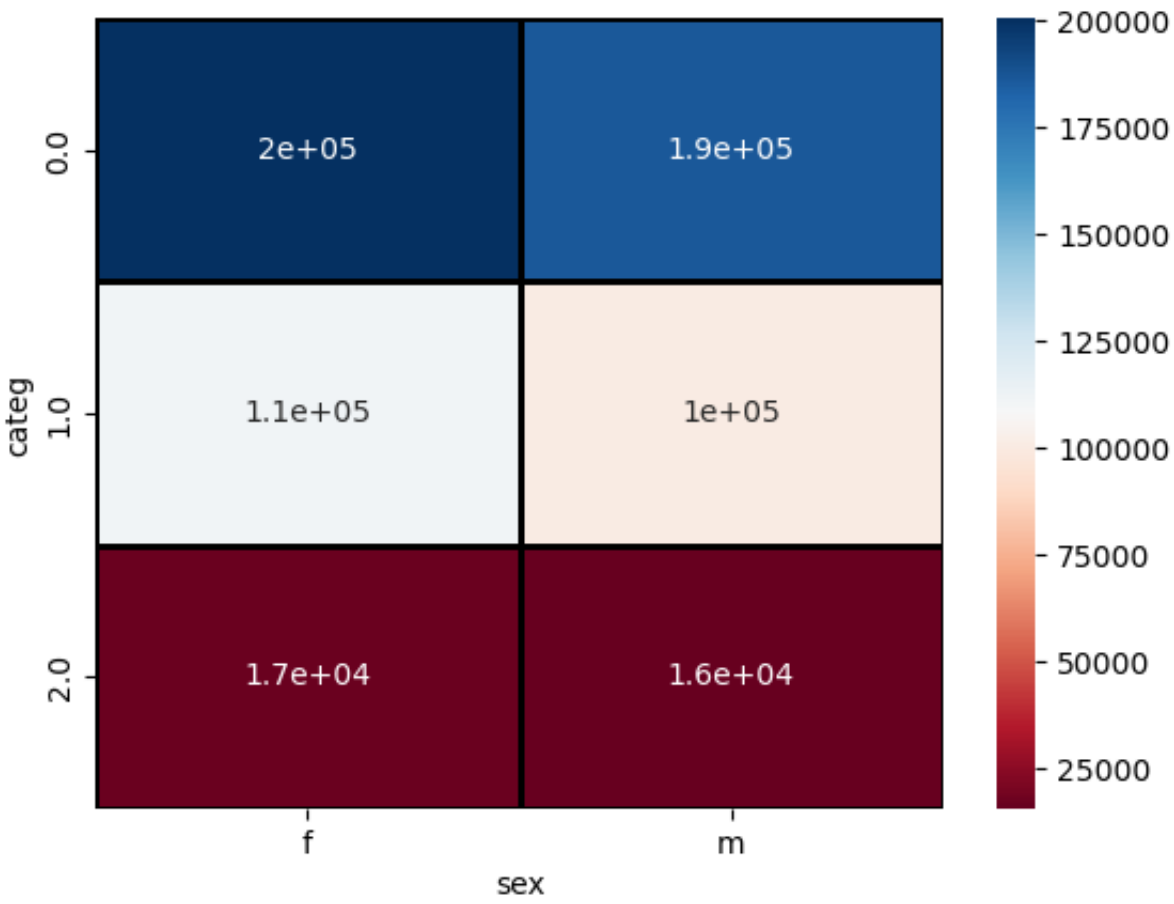
In [54]:
```
# Table de contingence normalisée

table, results = rp.crosstab(
    perso['categ'], perso['sex'], prop='col', test='chi-square', margins=
table
```

Out[54]:

sex	f	m	All
categ			
0.0	61.03	61.50	61.25
1.0	33.82	33.28	33.56
2.0	5.16	5.22	5.18
All	100.00	100.00	100.00

In [55]:
```
# Heatmap

sns.heatmap(table_contingence,annot=True, cmap='RdBu', linewidths=1, line
```

Out[55]:
```
<AxesSubplot:xlabel='sex', ylabel='categ'>
```



In [56]:

```
# Test de Chi 2

stat, p, dof, expected = sts.chi2_contingency(table_contingence)


resultats_test =sts.chi2_contingency(table_contingence)

print ("Statistique de test :", resultats_test [0],'\n')
print ("P valeur :", resultats_test [1],'\n')
print ("Degré de liberté :", resultats_test [2],'\n')

if p > 0.5:
    print("On accepte H0: Il n'y a pas de lien entre le genre et la catég
else:
    print("On rejette H0: Il y'a un lien entre le genre et la catégorie d
```

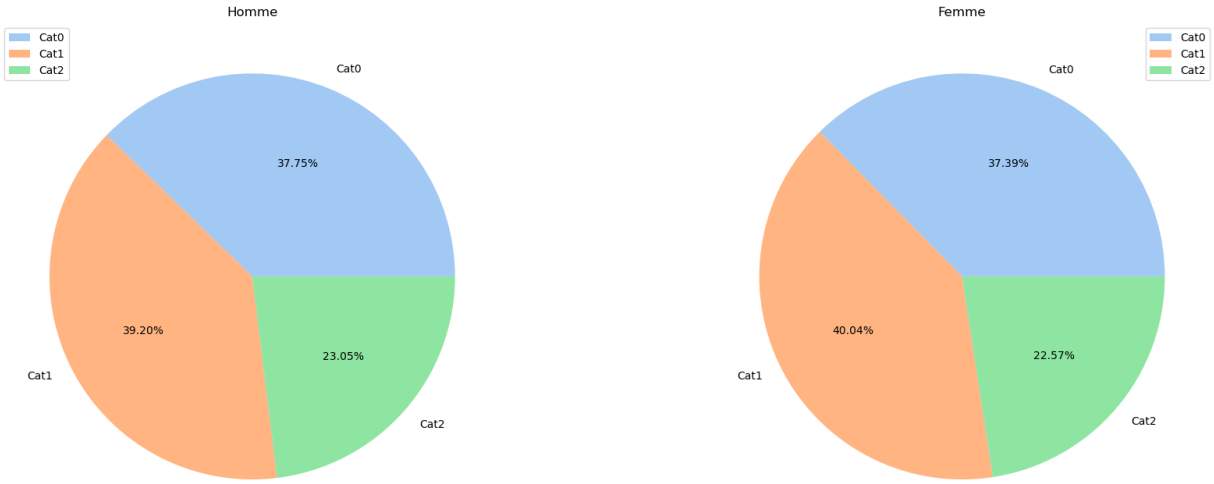
```
Statistique de test : 20.296119511850566

P valeur : 3.915197241110953e-05

Degré de liberté : 2

On rejette H0: Il y'a un lien entre le genre et la catégorie de livre ach
etée
```

In [57]:
```
# Test V de Crameur pour mésurer l'intensité entre sexe et catégorie

# Test de cramer

def cramers(table):
    chi2 = sts.chi2_contingency(table)[0]
    n = sum(table.sum())
    return np.sqrt(chi2 / (n*(min(table.shape)-1)))


result = cramers(table_contingence)

print("V de Crameur =", result,'\n')

if result <= 0.1:
    print("L'intensité entre les deux variables est très faible")
elif result <= 0.2:
    print("L'intensité entre les deux variables est faible")
elif result <= 0.5:
    print("L'intensité entre les deux variables est moyenne")
elif result >= 0.5:
    print("L'intensité entre les deux variables est forte")
```

```
V de Crameur = 0.00566730818383317

L'intensité entre les deux variables est très faible
```

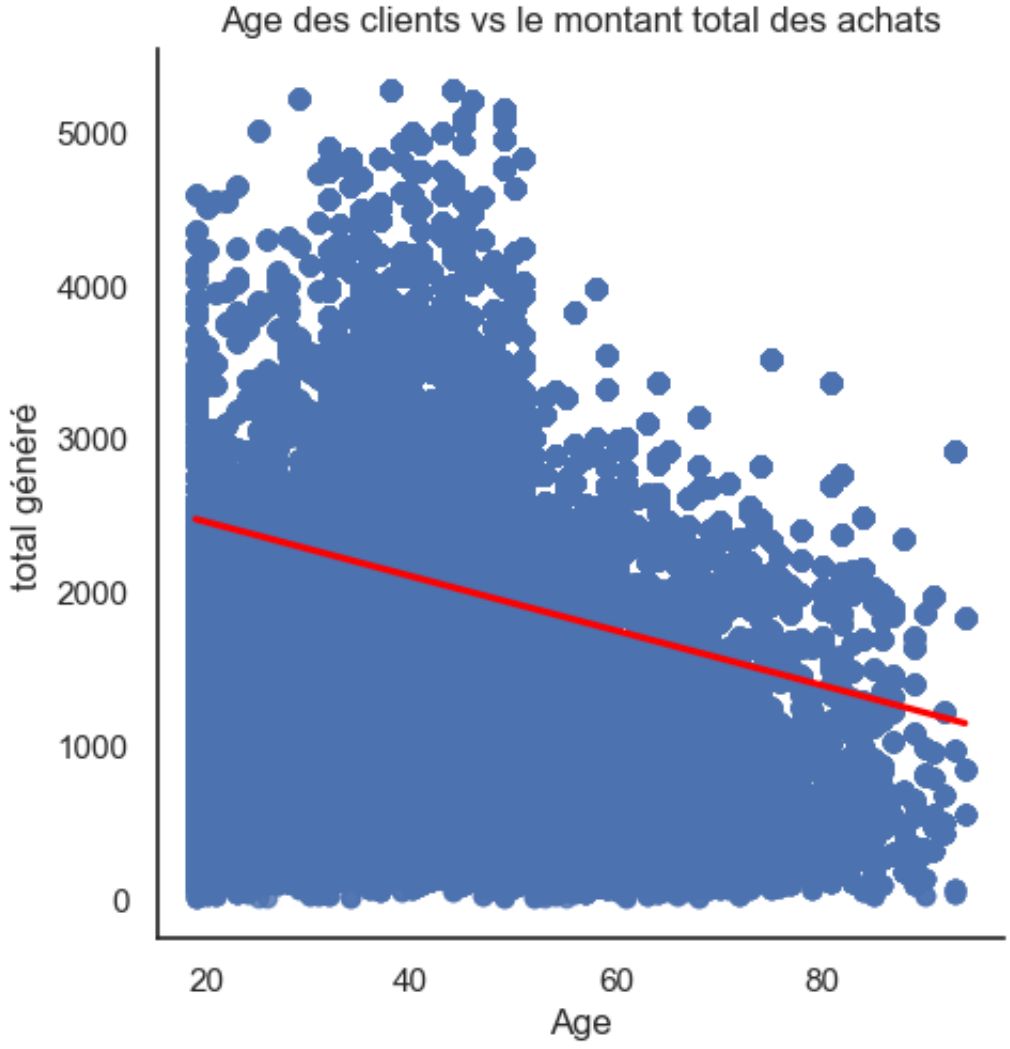
In [58]:
```
# Calcul du chiffres d'affaire par catégorie de sexe

# Création d'un mask pour isolé le sexe

sex_f = perso.loc[data_lapage['sex'] == 'f']
sex_m = perso.loc[data_lapage['sex'] == 'm']

# Calcul du chiffres d'affaire

CA_f = sex_f.groupby(['categ']).sum('price')
CA_f = CA_f[['price']].reset_index()

CA_m = sex_m.groupby(['categ']).sum('price')
CA_m = CA_m[['price']].reset_index()
```

In [59]:
```
# Repartition de CA par catégorie de sexe

plt.figure(figsize=(20, 7))
plt.subplot(121)
plt.pie(CA_m['price'], labels=['Cat0', 'Cat1', 'Cat2'], autopct='%.2f%%')
plt.title("Homme")
sns.set_palette('pastel')
plt.legend(loc='upper left')
plt.subplot(122)
plt.pie(CA_f['price'], labels=['Cat0', 'Cat1', 'Cat2'], autopct='%.2f%%')
plt.title("Femme")
sns.set_palette('pastel')
plt.legend(loc='upper right')
plt.tight_layout()
plt.show()
```



8. Lien entre l'âge des clients et le montant total des achats

In [60]:
```
# Aperçu du lien

sns.set(font_scale=1, style="white")
ax = sns.lmplot(x='age', y='total_genere',
                data=perso, line_kws={'color': 'red'})
ax.set(xlabel='Age', ylabel='total généré')
plt.title("Age des clients vs le montant total des achats")
plt.show()
```



Hypothèses de test :

H0 : Il n'y a pas de lien entre l'Age et le montant total des achats

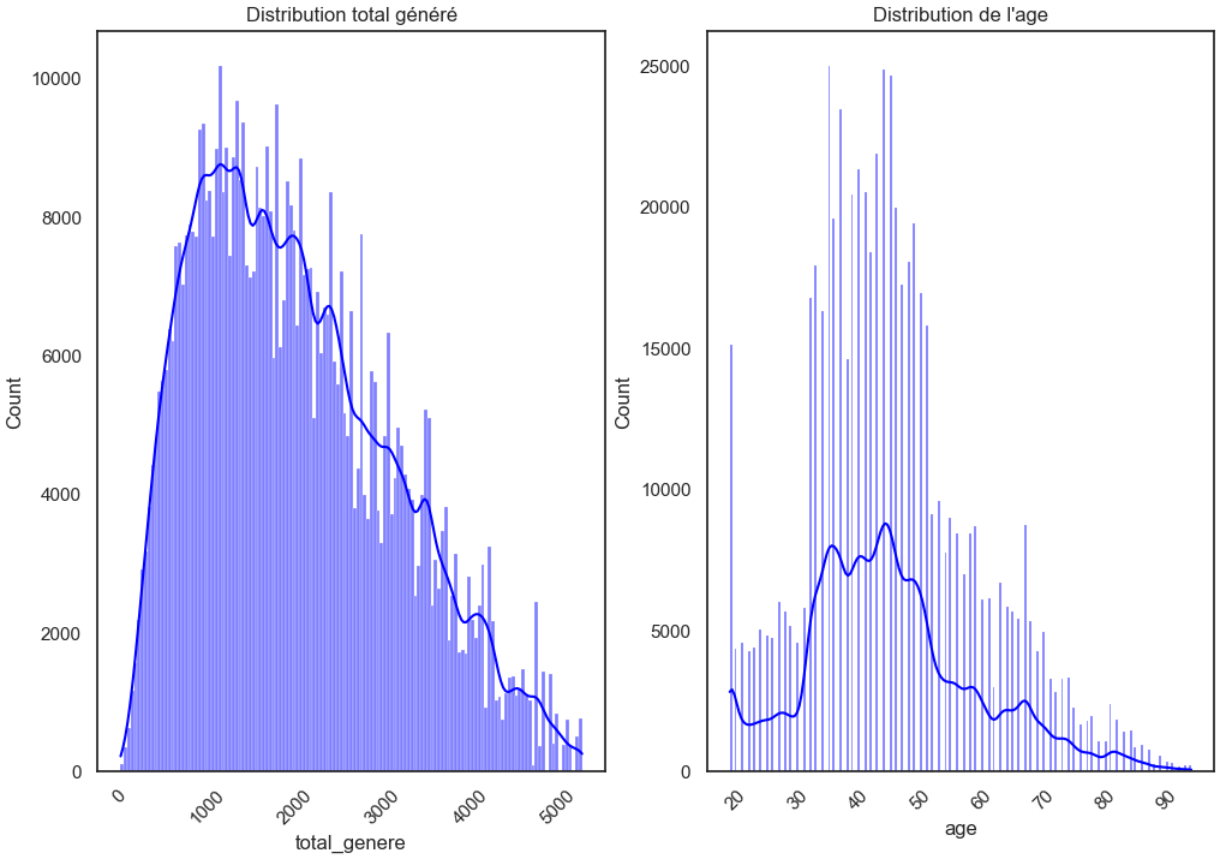
H1 : On rejette H0, il y a un lien entre l'Age et le montant total des achats

Seuil α = 0.05

Si pvalue > α, on accepte H0

Si pvalue < α, on rejette H0

In [61]:
```
# Distribution Total généré et Age

plt.figure(figsize=(12, 8))

plt.subplot(121)
sns.histplot(data=perso, x='total_genere', kde=True, color='blue')
plt.xticks(rotation=45)
plt.title('Distribution total généré')

plt.subplot(122)
sns.histplot(data=perso, x='age', kde=True, color='blue')
plt.xticks(rotation=45)
plt.title("Distribution de l'age")

plt.show()
```



Vérifions si nos variables suivent une distribution Gaussienne

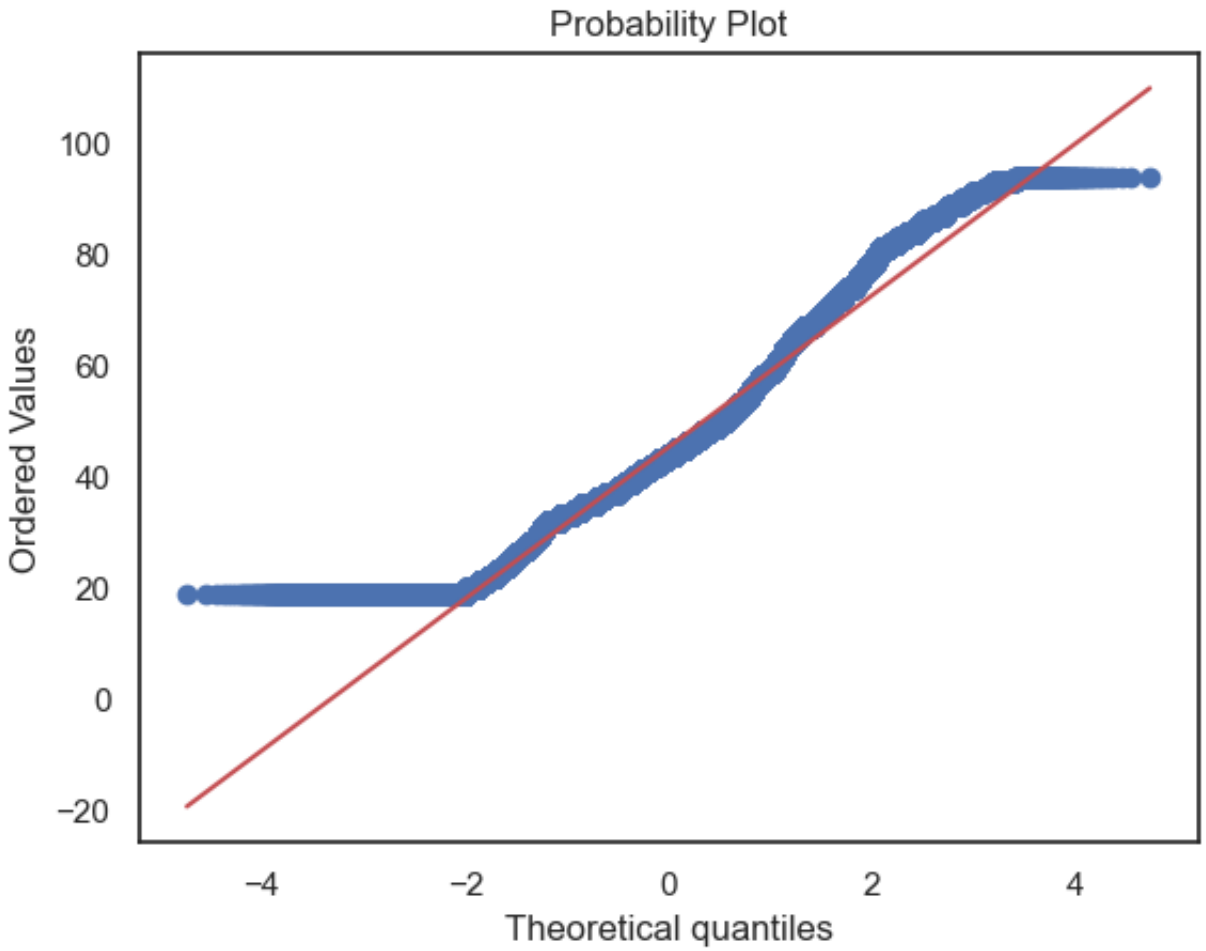
Hypothèses:

H0: Les variables suivent une distribution Gaussienne

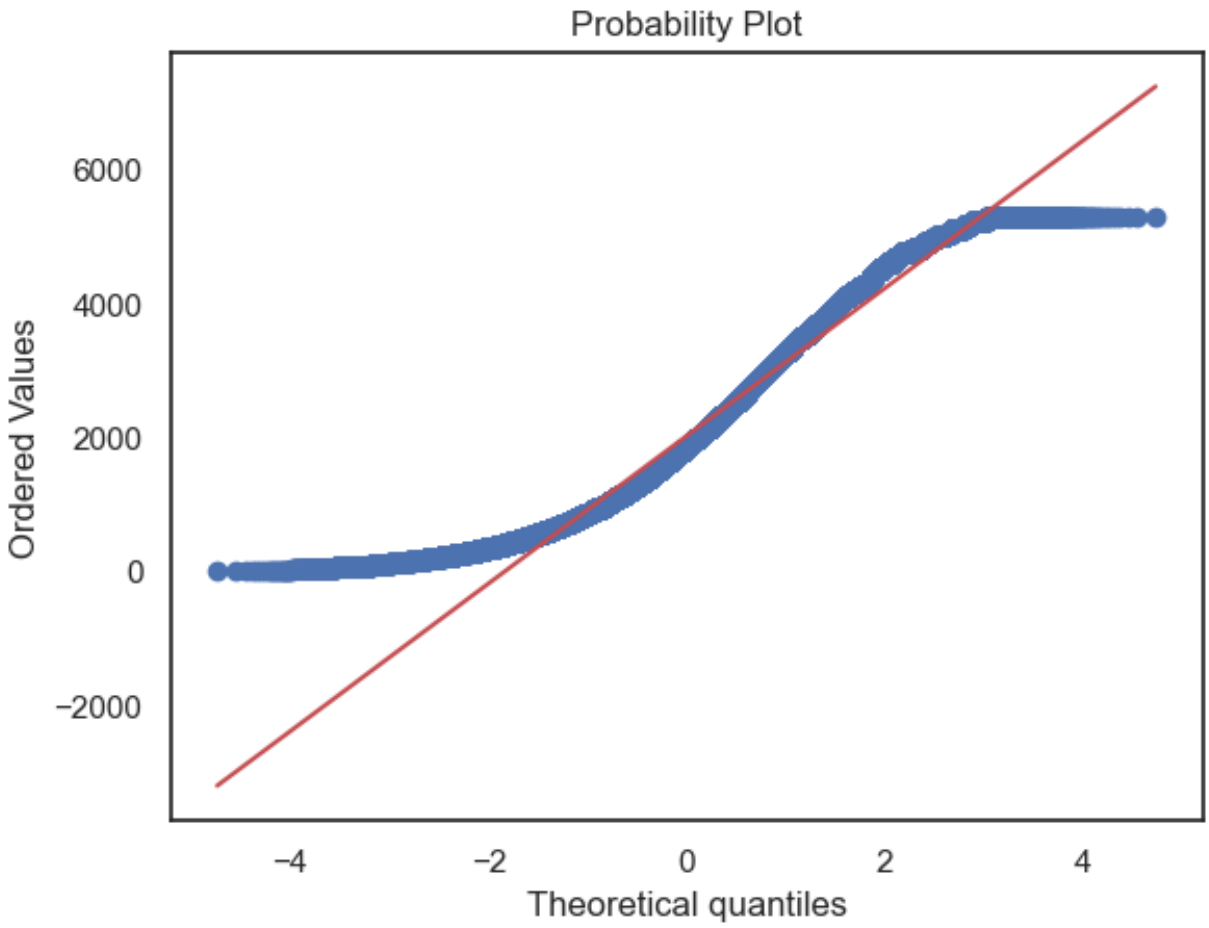
H1: Les variables ne suivent pas une distribution Gaussienne

Seuil alpha α = 0.05

In [62]:
```
# Graphique de probabilité par la variable Age

sts.probplot(perso['age'], dist="norm", plot=pylab)

pylab.show()
```



In [63]:
```
# Graphique de probabilité par la variable total généré

sts.probplot(perso['total_genere'], dist="norm", plot=pylab)

pylab.show()
```



Nous voyons graphiquement que nos deux variables ne suivent pas la loi normale, nous allons confirmer cela a partir du test de Shapiro et Kolmogorov-Smirnov

In [64]:
```
#Test de Shapiro sur la variable âge

rng = perso['age']
x = sts.norm.rvs(loc=5, scale=3, size=5000)
shapiro_test = sts.shapiro(x)

shapiro_test

if p >0.5:
    print("On accepte H0: La variable suit une distribution Gaussienne")
else:
    print("On rejette H0: La variable ne suit pas une distribution Gaussi
```

```
On rejette H0: La variable ne suit pas une distribution Gaussiennee
```

In [65]:
```
#Test de Shapiro sur la variable total généré

rng = perso['total_genere']
x = sts.norm.rvs(loc=5, scale=3, size=5000)
shapiro_test = sts.shapiro(x)

shapiro_test

if p >0.5:
    print("On accepte H0: La variable suit une distribution Gaussienne")
else:
    print("On rejette H0: La variable ne suit pas une distribution Gaussi
```

```
On rejette H0: La variable ne suit pas une distribution Gaussiennee
```

In [66]:
```
# Test de Kolmogorov-Smirnov sur la variable age

# estimation du parametre par methode du maximum de Vraisemblance
lbda = 1/perso['age'].mean()

stat_test, p_value = kstest(perso['age'], 'expon', args=(0, lbda))

print("la statistique de test:", stat_test, '\n')
print("la p_value du test:", p_value, '\n')

if p > 0.5:
    print("On accepte H0: La variable suit une distribution Gaussienne")
else:
    print("On rejette H0: La variable ne suit pas une distribution Gaussi
```

```
la statistique de test: 1.0

la p_value du test: 0.0

On rejette H0: La variable ne suit pas une distribution Gaussiennee
```

In [67]:
```
#Test de Kolmogorov-Smirnov sur la variable total généré

# estimation du parametre par methode du maximum de Vraisemblance
lbda = 1/perso['total_genere'].mean()

stat_test, p_value = kstest(perso['total_genere'], 'expon', args=(0, lbda

print("la statistique de test:", stat_test, '\n')
print("la p_value du test:", p_value, '\n')
if p > 0.5:
    print("On accepte H0: La variable suit une distribution Gaussienne")
else:
    print("On rejette H0: La variable ne suit pas une distribution Gaussi
```

```
la statistique de test: 1.0

la p_value du test: 0.0

On rejette H0: La variable ne suit pas une distribution Gaussiennee
```

Nos variables ne suivant pas une distribution Gaussienne, nous allons donc utiliser un test non-paramétrique afin de voir s'il y'a corrélation ou non entre l'Age des clients et le montant total des achats

In [68]:
```
# Test de Spearman

sts.spearmanr(perso['age'], perso['total_genere'])

print(sts.spearmanr(perso['age'], perso['total_genere']), '\n')
if p > 0.5:
    print("Il n'y a pas de lien entre l'Age et le montant total des achat
else:
    print("On rejette H0, il y a un lien entre l'Age et le montant total
```

```
SpearmanrResult(correlation=-0.20615575752998558, pvalue=0.0)

On rejette H0, il y a un lien entre l'Age et le montant total des achats
```

Plus l'age des clients augmente, plus le total généré par ces derniers diminue

8.1. Lien entre la fréquence d'achat et l'âge des clients

Nous sommes face à deux variables quantitatives: Continue pour la fréquence d'achat et Discrète pour l'âge

Hypothèses de test :

H0 : Il n'y a pas de lien entre l'Age des clients et la fréquence d'achat

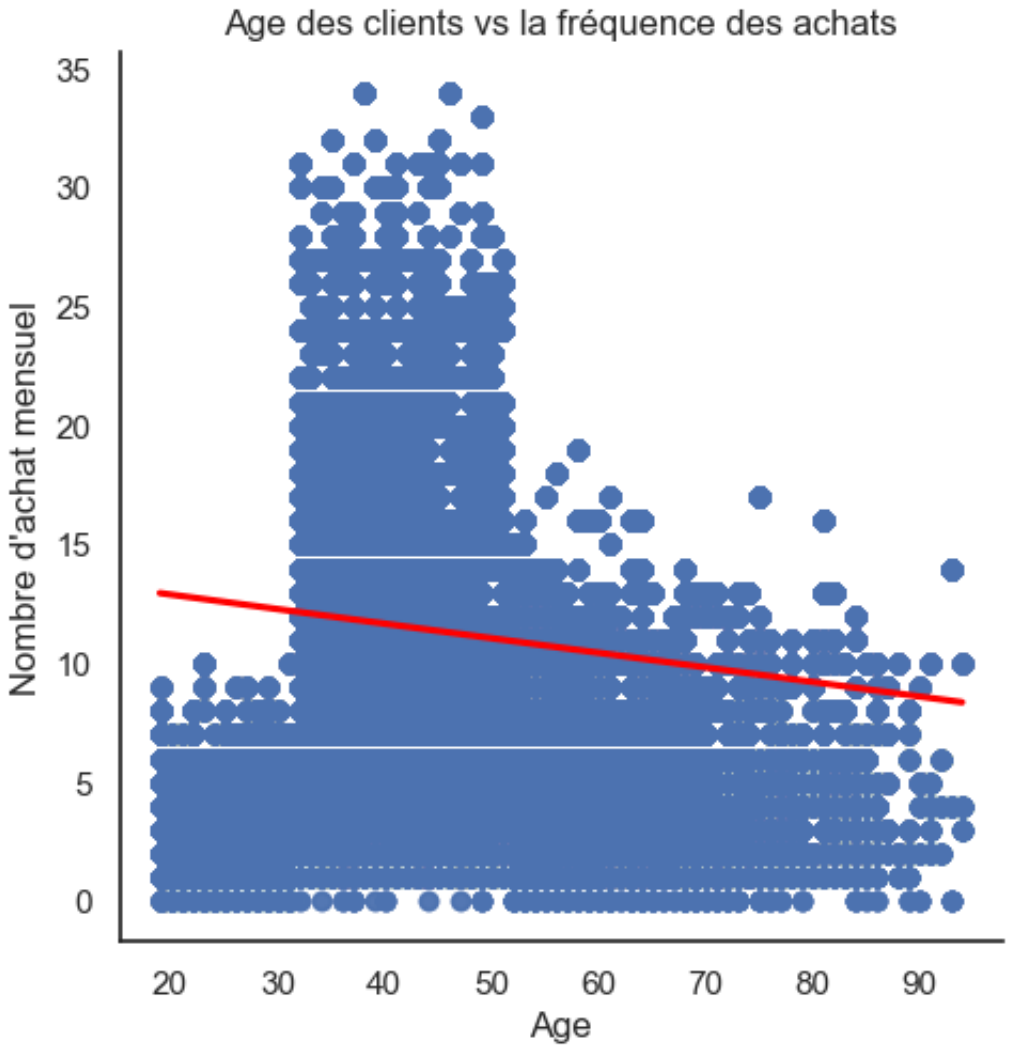
H1 : On rejette H0, il y a un lien entre l'Age des clients et la fréquence d'achat

Seuil α = 0.05

Si pvalue > α, on accepte H0

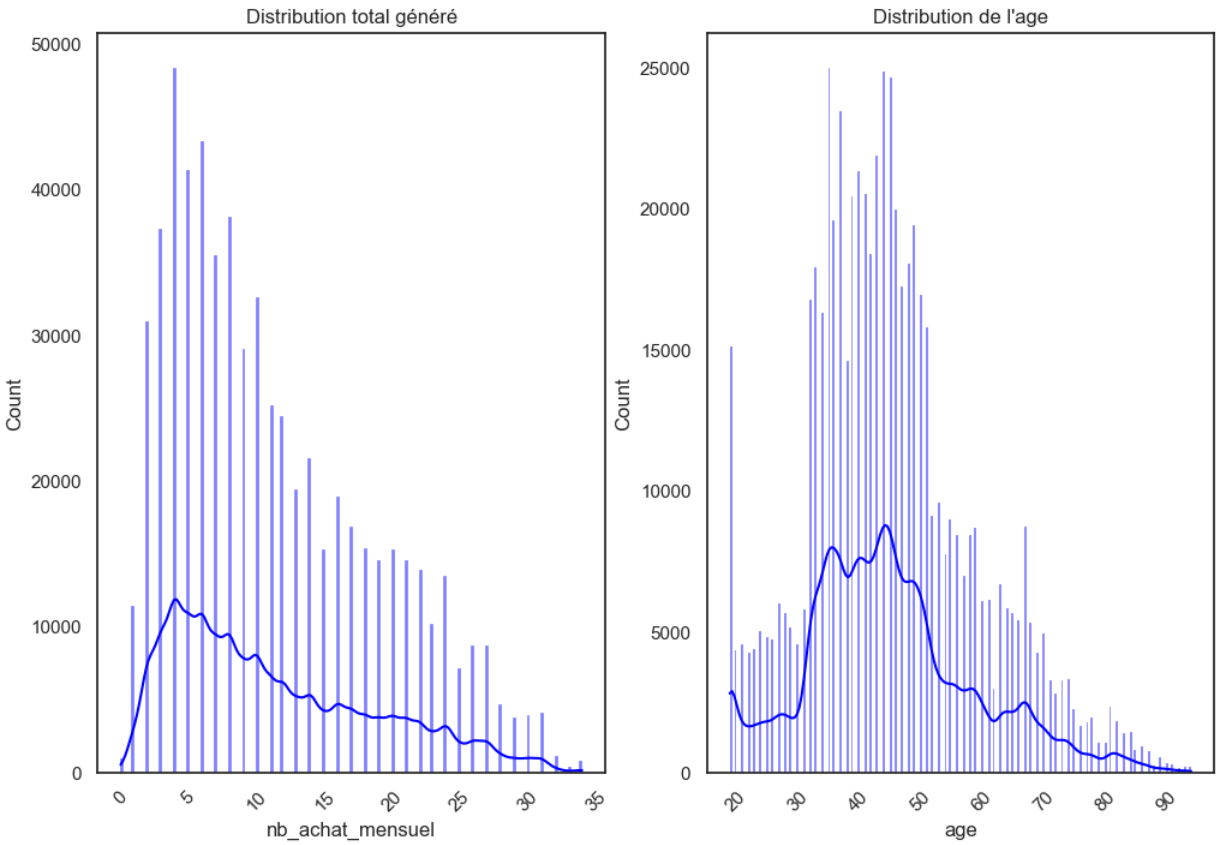
Si pvalue < α, on rejette H0

In [69]:
```
# Aperçu du lien

sns.set(font_scale=1, style="white")
ax = sns.lmplot(x='age', y='nb_achat_mensuel',
                data=perso, line_kws={'color': 'red'})
ax.set(xlabel='Age', ylabel="Nombre d'achat mensuel")
plt.title("Age des clients vs la fréquence des achats")
plt.show()
```



In [70]:

```
# Distribution Total fréquence d'achat et Age

plt.figure(figsize=(12, 8))

plt.subplot(121)
sns.histplot(data=perso, x='nb_achat_mensuel', kde=True, color='blue')
plt.xticks(rotation=45)
plt.title('Distribution total généré')

plt.subplot(122)
sns.histplot(data=perso, x='age', kde=True, color='blue')
plt.xticks(rotation=45)
plt.title("Distribution de l'age")

plt.show()
```



Vérifions si nos variables suivent une distribution Gaussienne

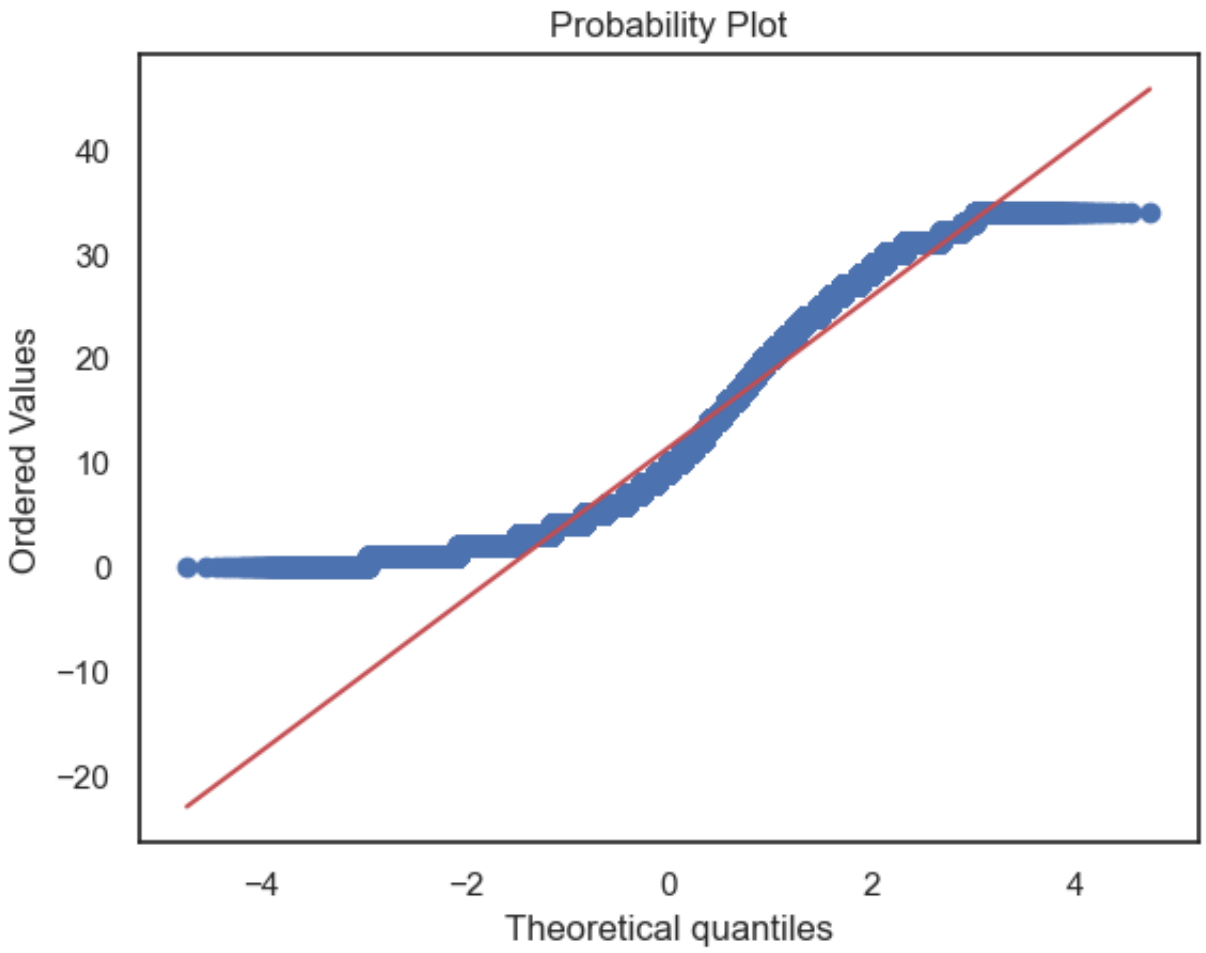
Hypothèses:

H0: Les variables suivent une distribution Gaussienne

H1: Les variables ne suivent pas une distribution Gaussienne

Seuil alpha α = 0.05

In [71]:
```
# Graphique de probabilité pour la variable nb_achat_mensuel

sts.probplot(perso['nb_achat_mensuel'], dist="norm", plot=pylab)

pylab.show()
```



Nous voyons graphiquement que nos deux variables ne suivent pas la loi normale, nous allons confirmer cela a partir du test de Shapiro et Kolmogorov-Smirnov

In [72]:
```
#Test de Shapiro sur la variable total généré

rng = perso['nb_achat_mensuel']
x = sts.norm.rvs(loc=5, scale=3, size=5000)
shapiro_test = sts.shapiro(x)

shapiro_test

if p >0.5:
    print("On accepte H0: La variable suit une distribution Gaussienne")
else:
    print("On rejette H0: La variable ne suit pas une distribution Gaussi
```

```
On rejette H0: La variable ne suit pas une distribution Gaussiennee
```

In [73]:
```
# Test de Kolmogorov-Smirnov sur la variable nb_achat_mensuel

# estimation du parametre par methode du maximum de Vraisemblance
lbda = 1/perso['nb_achat_mensuel'].mean()

stat_test, p_value = kstest(perso['nb_achat_mensuel'], 'expon', args=(0,

print("la statistique de test:", stat_test, '\n')
print("la p_value du test:", p_value, '\n')

if p > 0.5:
    print("On accepte H0: La variable suit une distribution Gaussienne")
else:
    print("On rejette H0: La variable ne suit pas une distribution Gaussi
```

```
la statistique de test: 0.9984247171657712

la p_value du test: 0.0

On rejette H0: La variable ne suit pas une distribution Gaussiennee
```

Nos variables ne suivant pas une distribution Gaussienne, nous allons donc utiliser un test non-paramétrique afin de voir s'il y'a corrélation ou non entre l'Age des clients et la fréquence d'achat

In [74]:
```
# Test de Spearman

sts.spearmanr(perso['age'], perso['nb_achat_mensuel'])

print(sts.spearmanr(perso['age'], perso['nb_achat_mensuel']), '\n')
if p > 0.5:
    print("Il n'y a pas de lien entre l'Age des clients et la fréquence d
else:
    print("On rejette H0, il y a un lien entre l'Age des clients et la fr
```

```
SpearmanrResult(correlation=-0.05421263135048499, pvalue=0.0)

On rejette H0, il y a un lien entre l'Age des clients et la fréquence d'a
chat
```

Plus l'age des clients augmente, plus la fréquence d'achat de ces derniers diminue

8.2. Lien entre la taille du panier moyen et l'age des clients

Nous sommes face à deux variables quantitatives: Continue pour le Panier Moyen et Discrète pour l'âge

Hypothèses de test :

H0 : Il n'y a pas de lien entre l'Age des clients et le panier moyen

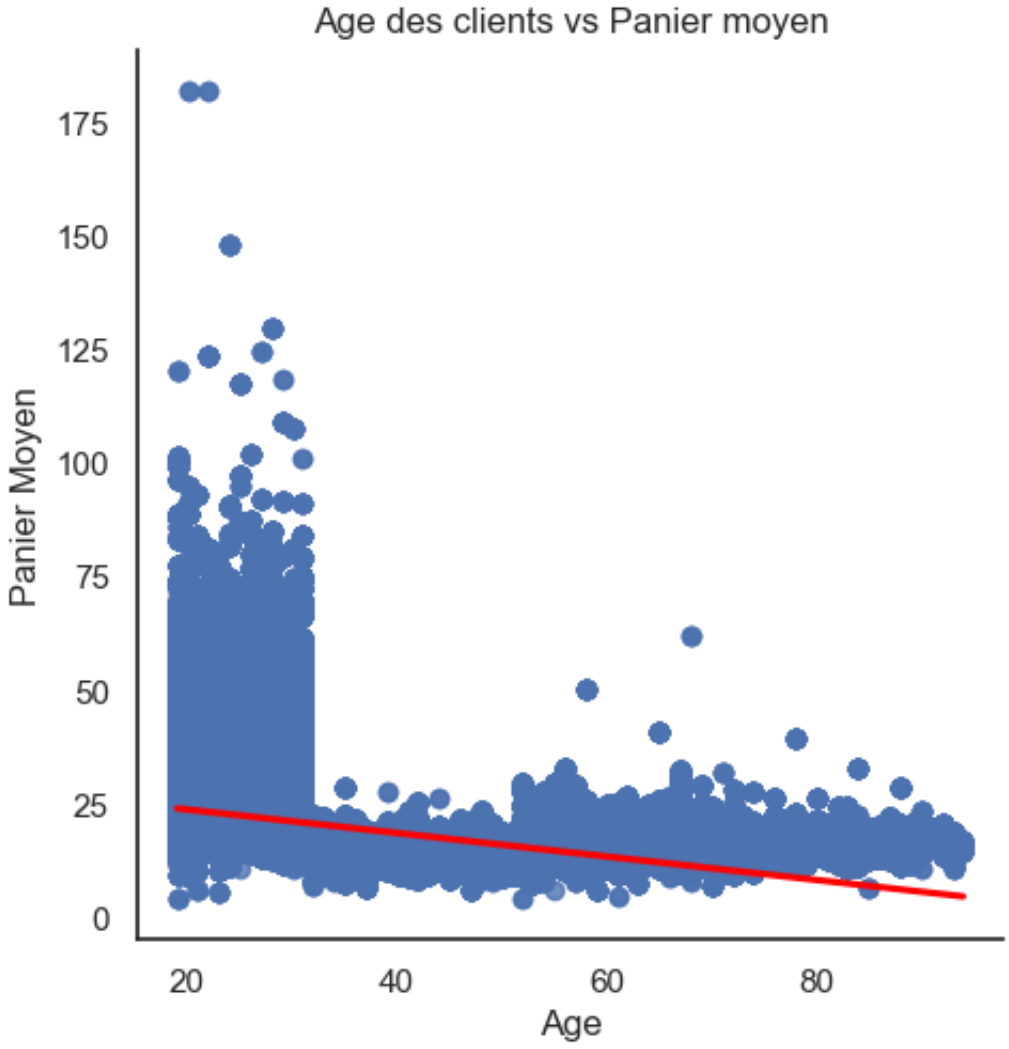
H1 : On rejette H0, il y a un lien entre l'Age des clients et le panier moyen

Seuil α = 0.05

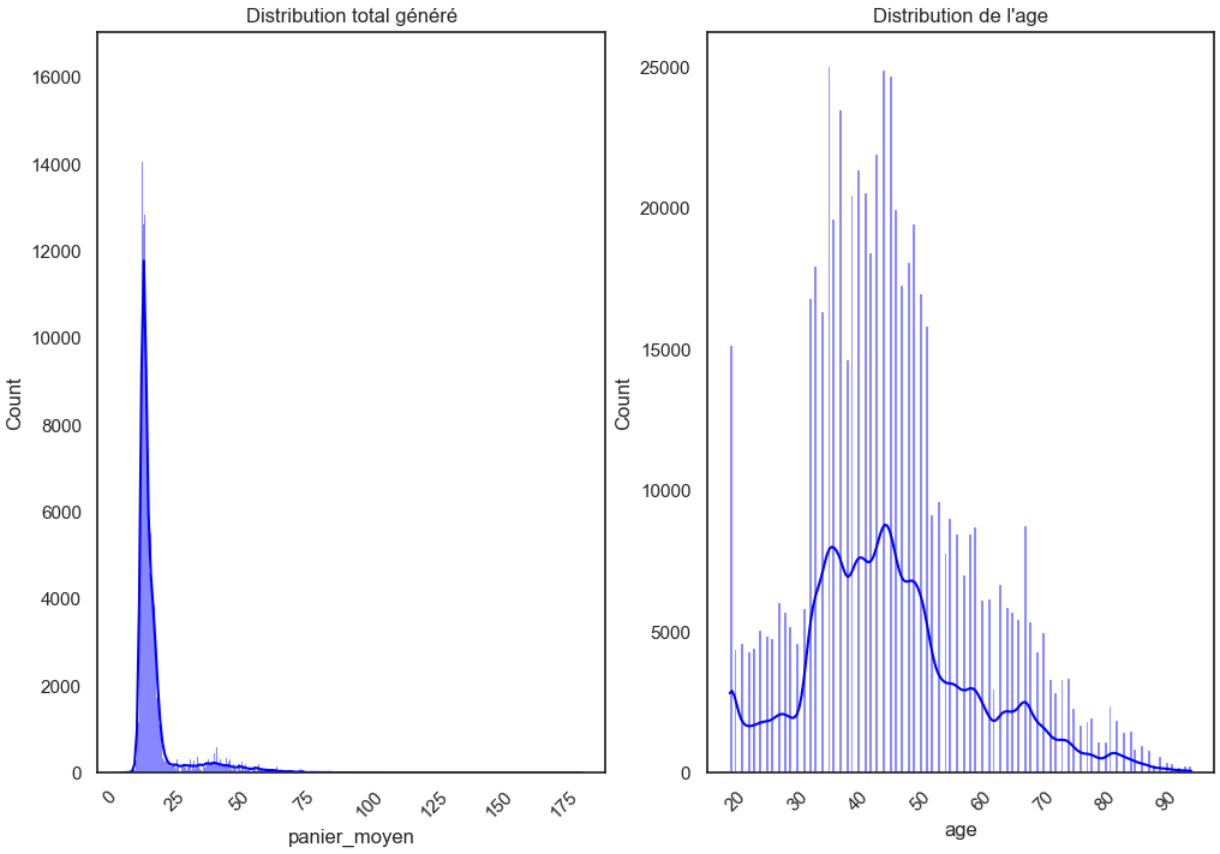
Si pvalue > α, on accepte H0

Si pvalue < α, on rejette H0

In [75]:
```
# Aperçu du lien

sns.set(font_scale=1,style="white")
ax =sns.lmplot(x='age',y='panier_moyen', data=perso,line_kws={'color': 'r
ax.set(xlabel='Age', ylabel='Panier Moyen')
plt.title("Age des clients vs Panier moyen");
plt.show()
```



In [76]:
```
# Distribution panier_moyen et Age

plt.figure(figsize=(12, 8))

plt.subplot(121)
sns.histplot(data=perso, x='panier_moyen', kde=True, color='blue')
plt.xticks(rotation=45)
plt.title('Distribution total généré')

plt.subplot(122)
sns.histplot(data=perso, x='age', kde=True, color='blue')
plt.xticks(rotation=45)
plt.title("Distribution de l'age")

plt.show()
```



Vérifions si nos variables suivent une distribution Gaussienne

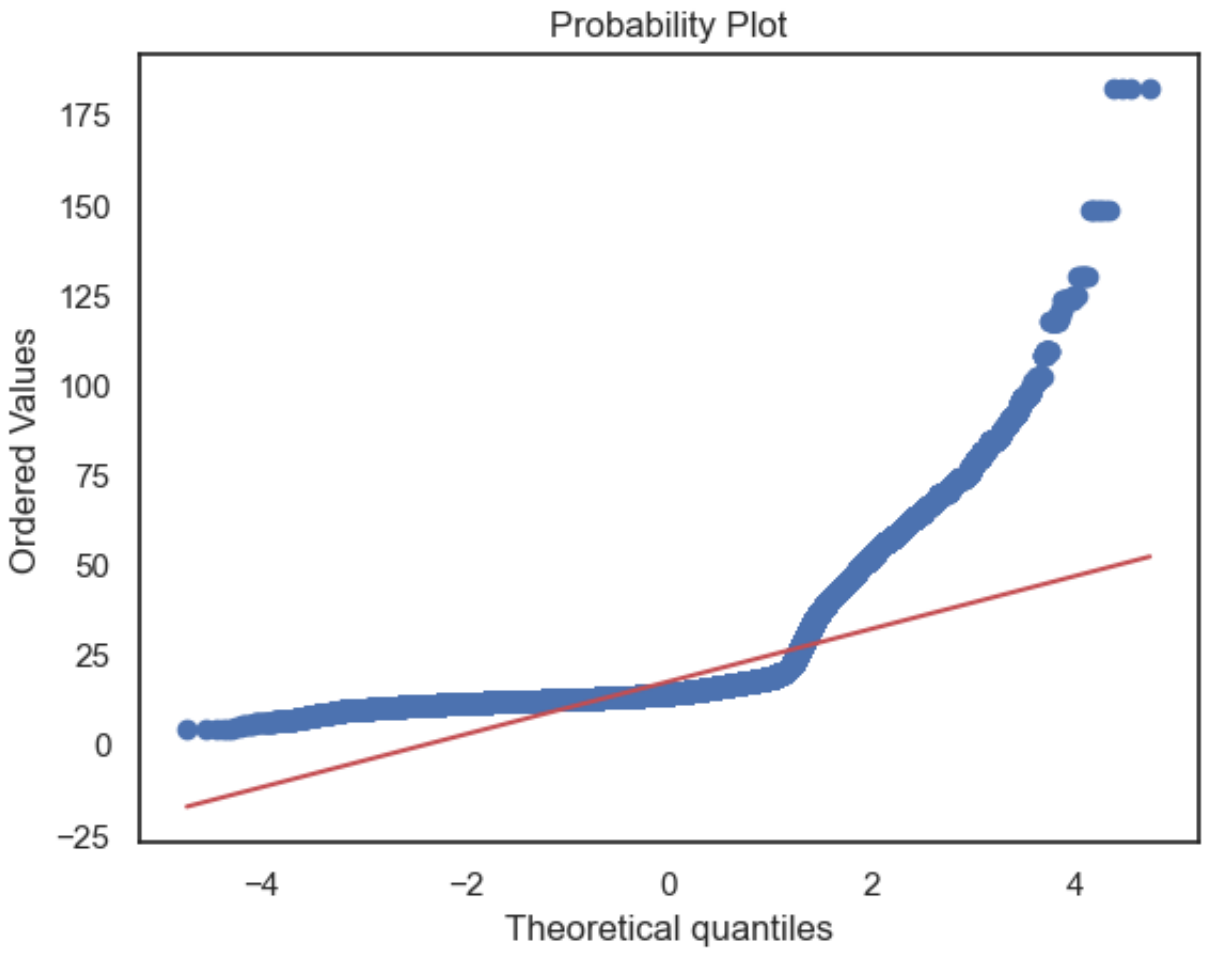
Hypothèses:

H0: Les variables suivent une distribution Gaussienne

H1: Les variables ne suivent pas une distribution Gaussienne

Seuil alpha α = 0.05

In [77]:
```
# Graphique de probabilité par la variable panier_moyen

sts.probplot(perso['panier_moyen'], dist="norm", plot=pylab)

pylab.show()
```



Nous voyons graphiquement que nos deux variables ne suivent pas la loi normale, nous allons confirmer cela a partir du test de Shapiro et Kolmogorov-Smirnov

In [78]:
```
#Test de Shapiro sur la variable panier_moyen

rng = perso['panier_moyen']
x = sts.norm.rvs(loc=5, scale=3, size=5000)
shapiro_test = sts.shapiro(x)

shapiro_test

if p >0.5:
    print("On accepte H0: La variable suit une distribution Gaussienne")
else:
    print("On rejette H0: La variable ne suit pas une distribution Gaussi
```

```
On rejette H0: La variable ne suit pas une distribution Gaussiennee
```

In [79]:
```
# Test de Kolmogorov-Smirnov sur la variable panier_moyen

# estimation du parametre par methode du maximum de Vraisemblance
lbda = 1/perso['panier_moyen'].mean()

stat_test, p_value = kstest(perso['panier_moyen'], 'expon', args=(0, lbda

print("la statistique de test:", stat_test, '\n')
print("la p_value du test est:", p_value, '\n')

if p > 0.5:
    print("On accepte H0: La variable suit une distribution Gaussienne")
else:
    print("On rejette H0: La variable ne suit pas une distribution Gaussi
```

```
la statistique de test: 1.0

la p_value du test est: 0.0

On rejette H0: La variable ne suit pas une distribution Gaussiennee
```

Nos variables ne suivant pas une distribution Gaussienne, nous allons donc utiliser un test non-paramétrique afin de voir s'il y'a corrélation ou non entre l'Age des clients et le panier moyen

In [80]:
```
# Test de Spearman

sts.spearmanr(perso['age'], perso['panier_moyen'])

print(sts.spearmanr(perso['age'], perso['panier_moyen']), '\n')

if p > 0.5:
    print("Il n'y a pas de lien entre l'Age des clients et le panier moye
else:
    print("On rejette H0, il y a un lien entre l'Age des clients et le pa
```

```
SpearmanrResult(correlation=0.06982009891727811, pvalue=0.0)

On rejette H0, il y a un lien entre l'Age des clients et le panier moyen
```

Plus l'age des clients augmente, plus le panier moyen de ces derniers diminue

8.3. Lien entre les catégories des livres achetés et l'âge des clients

Nous sommes face à une variable qualitative (Catégorie) et une variable quantitative discrète (Age)

Hypothèses de test :

H0 : Il n'y a pas de lien entre l'Age des clients et la catégorie de livre acheté

H1 : On rejette H0, il y a un lien entre l'Age des clients et la catégorie de livre acheté

Seuil α = 0.05

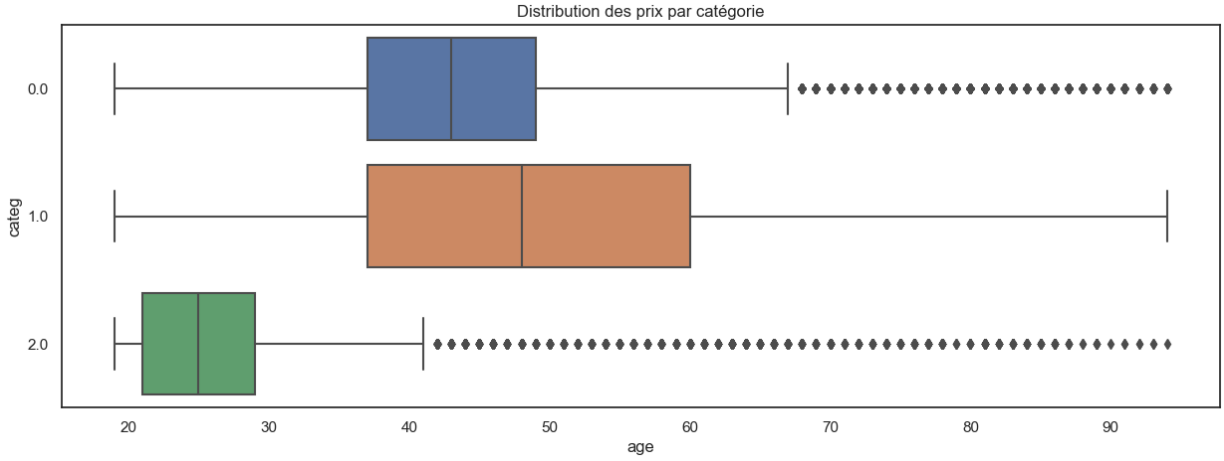
Si pvalue > α, on accepte H0

Si pvalue < α, on rejette H0

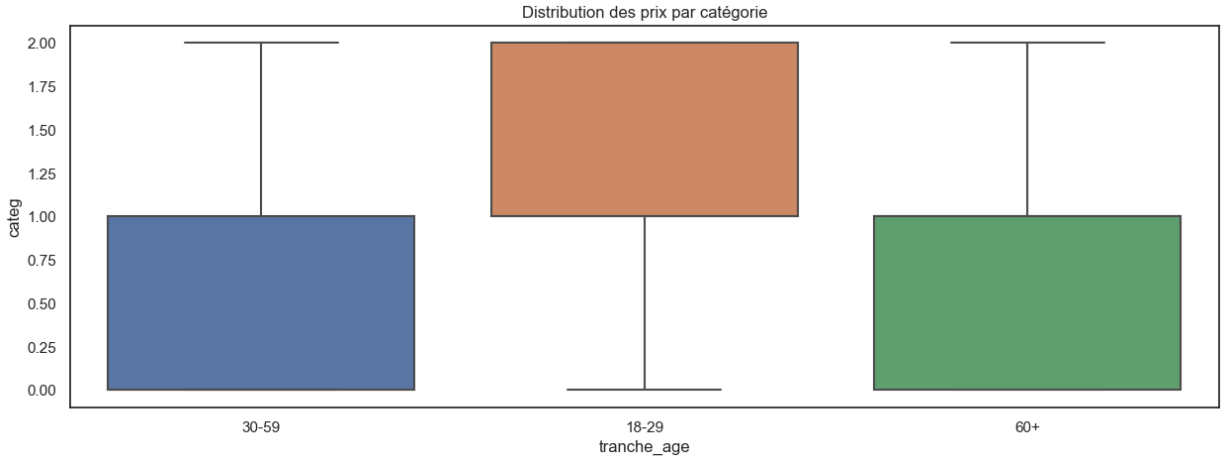
In [81]:
```
#Anova (F-TEST)

df_anova = perso[['categ', 'age','tranche_age']]
groupe = pd.unique(df_anova.categ.values)
print(groupe)
d_data = {grp:df_anova['age'][df_anova.categ ==grp] for grp in groupe}
print(d_data)
```

```
[0. 1. 2.]
{0.0: 0         37
4         43
5         51
6         42
7         38
          ..
678504    59
678505    61
678507    72
678509    35
678510    37
Name: age, Length: 387071, dtype: int64, 1.0: 1         35
9         50
11        63
12        65
20        59
          ..
678502    21
678503    40
678506    58
678508    46
678511    28
Name: age, Length: 212085, dtype: int64, 2.0: 3         23
18        21
33        19
39        19
55        29
          ..
678364    26
678388    29
678417    19
678421    19
678451    19
Name: age, Length: 32760, dtype: int64}
```

In [82]:
```
# Distribution des prix par catégorie

plt.figure(figsize=(15, 5))
sns.boxplot(data=df_anova, x=df_anova['age'], y=df_anova['categ'], orient
plt.title('Distribution des prix par catégorie')
plt.savefig('Distribution des prix par catégorie.jpeg', dpi=300, bbox_inc
plt.show();
```



In [83]:
```
plt.figure(figsize=(15, 5))
sns.boxplot(data=df_anova, x=df_anova['tranche_age'], y=df_anova['categ']
plt.title('Distribution des prix par catégorie')
plt.show();
```



In [84]:
```
#Anova (F-TEST)

F, p = f_oneway(d_data[0.], d_data[1.], d_data[2.])
print("La Pvalue est", p)

if p > 0.5:
    print("Il n'y a pas de lien entre l'Age des clients et la catégorie d
else:
    print("On rejette H0, il y a un lien entre l'Age des clients et la ca
```

```
La Pvalue est 0.0
On rejette H0, il y a un lien entre l'Age des clients et la catégorie de
livre acheté
```

In [85]:
```
# Test de Kruskal-Wallis

serie_01 = df_anova["categ"].loc[df_anova["age"] == 0.]
serie_02 = df_anova["categ"].loc[df_anova["age"] == 1.]
serie_03 = df_anova["categ"].loc[df_anova["age"] == 2.]

print("La Pvalue est", p)
if p > 0.5:
    print("Il n'y a pas de lien entre l'Age des clients et la catégorie d
else:
    print("On rejette H0, il y a un lien entre l'Age des clients et la ca
```

```
La Pvalue est 0.0
On rejette H0, il y a un lien entre l'Age des clients et la catégorie de
livre acheté
```

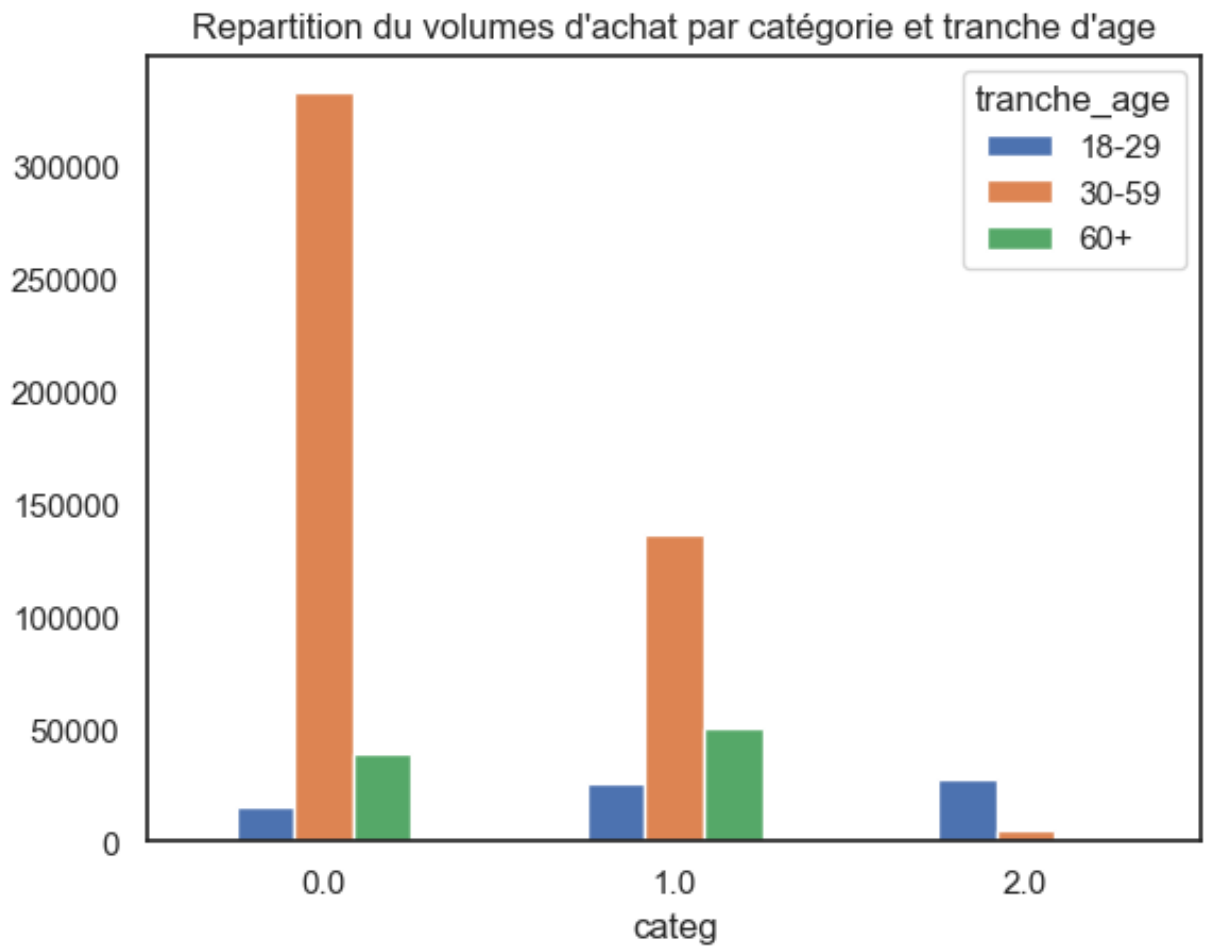
Lien entre les catégories des livres achetés et la tranche d'age des clients

In [86]:
```
# Création d'une table de contingence

table_contingence1 = pd.crosstab(
    perso['categ'], perso['tranche_age'])

table_contingence1
```

Out[86]:

tranche_age	18-29	30-59	60+
categ			
0.0	15430	332310	39331
1.0	25843	136339	49903
2.0	27449	4472	839

In [87]:
```
# Aperçu de la table de contingence

table_contingence1.plot.bar()
plt.title("Repartition du volumes d'achat par catégorie et tranche d'age"
plt.xticks(rotation=0);
```

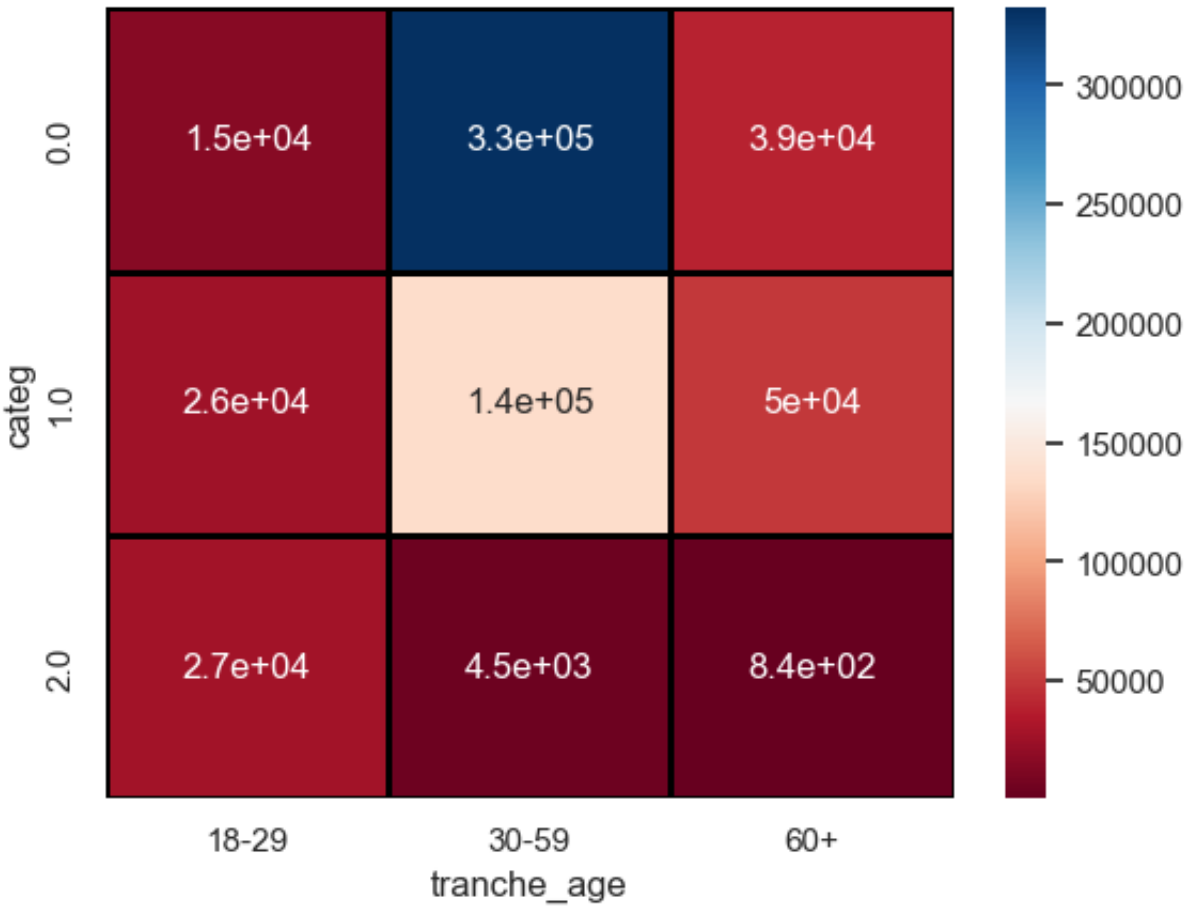


In [88]:
```
# Table de contingence normalisée

table1, results = rp.crosstab(
    perso['categ'], perso['tranche_age'], prop='col', test='chi-square',
table1
```

Out[88]:

tranche_age	18-29	30-59	60+	All
categ				
0.0	22.45	70.24	43.67	61.25
1.0	37.61	28.82	55.40	33.56
2.0	39.94	0.95	0.93	5.18
All	100.00	100.00	100.00	100.00

In [89]:
```
# Heatmap

sns.heatmap(table_contingence1,annot=True, cmap='RdBu', linewidths=1, lin
```



In [90]:
```
# Test de Chi 2

stat, p, dof, expected = sts.chi2_contingency(table_contingence1)

resultats_test =sts.chi2_contingency(table_contingence1)

print ("Statistique de test :", resultats_test [0],'\n')
print ("P valeur :", resultats_test [1],'\n')
print ("Degré de liberté :", resultats_test [2],'\n')

if p > 0.5:
    print("On accepte H0: Il n'y a pas de lien entre la tranche d'age des
else:
    print("On rejette H0, il y a un lien entre la tranche d'age des clien
```

```
Statistique de test : 223672.54530228436

P valeur : 0.0

Degré de liberté : 4

On rejette H0, il y a un lien entre la tranche d'age des clients et la ca
tégorie de livre acheté
```

In [91]:
```
# Test de Kruskal-Wallis

serie_01 = df_anova["categ"].loc[df_anova["tranche_age"] == 0.]
serie_02 = df_anova["categ"].loc[df_anova["tranche_age"] == 1.]
serie_03 = df_anova["categ"].loc[df_anova["tranche_age"] == 2.]

print("La Pvalue est", p)
if p > 0.5:
    print("Il n'y a pas de lien entre la tranche d'age des clients et la
else:
    print("On rejette H0, il y a un lien entre la tranche d'age des clien
```

```
La Pvalue est 0.0
On rejette H0, il y a un lien entre la tranche d'age des clients et la ca
tégorie de livre acheté
```

In [92]:
```
# Test V de Crameur pour mésurer l'intensité entre sexe et catégorie

# Test de cramer

def cramers(table):
    chi2 = sts.chi2_contingency(table)[0]
    n = sum(table.sum())
    return np.sqrt(chi2 / (n*(min(table.shape)-1)))


result = cramers(table_contingence1)

print("V de Crameur =", result, '\n')


if result <= 0.1:
    print("L'intensité entre les deux variables est très faible")
elif result <= 0.2:
    print("L'intensité entre les deux variables est faible")
elif result <= 0.5:
    print("L'intensité entre les deux variables est moyenne")
elif result >= 0.5:
    print("L'intensité entre les deux variables est forte")
```

```
V de Crameur = 0.42068949383637505

L'intensité entre les deux variables est moyenne
```

Les clients dont l'age vari entre 18-29 ans achètent plus la catégorie 2 et aussi la catégorie 1

Les clients dont l'age vari entre 30-59 ans achètent plus la catégorie 0

Les clients dont l'age est 60 et plus, ils achètent plus la catégorie 1

9. Conclusion

Aux termes de nos différents tests, études et analysent, il a été observé une tendance globale positive pour notre chiffre d'affaires (du 01-03-2021 au 28-02-2023)

L'année 2022 ayant été bien meilleure que celle de 2021, et vu la tendance observée, 2023 devraient être encore meilleure sinon garder les mêmes perspectives que 2022.

On note également un fort impact de l'âge sur le comportement de consommation de nos clients.

En effet, plus ils sont jeunes, plus ils achètent des livres et cela se ressent également dans notre chiffre d'affaires.

Si nous voulons rester sur cette lancée, est plus que nécessaire de fidéliser nos différents clients, mais surtout être à l'écoute de leurs besoins afin de fournir au mieux des produits qui répondent à leurs attentes.

↥