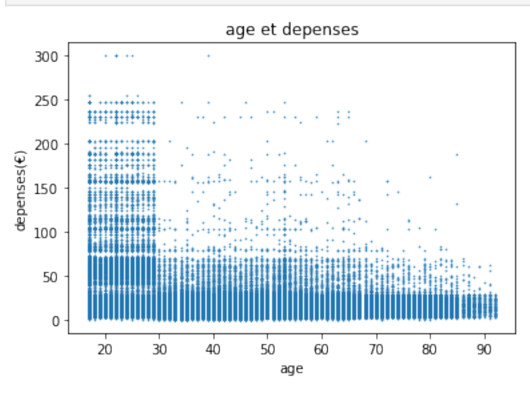
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
from datetime import datetime
import seaborn as sns
from scipy import stats as st
import statsmodels.formula.api
import statsmodels.api
```

In [2]:
```
#importation du fichier
ctp = pd.read_csv("ctp_ssOct.csv")
```

In [3]:
```
#Je regarde en premier les depenses en fonction de l'age avec un scatter plot
x = ctp["age"]
y = ctp["price"]
plt.title('age et depenses')
plt.xlabel('age')
plt.ylabel('depenses(€)')



plt.scatter(x, y, marker = '.', s = 1) #'.' points, s=1 taille des points
plt.savefig("scatter_depenses_age.png")
plt.show()
```



Il semble que le groupe jusqu'a 30 ans achete des produits plus cher

In [4]:
```
#help(plt.scatter)
```

In [5]:
```
#Dans la suite je regarde si il y a des correleations des differents variables prix, categorie et age
#avec une matrice de correleation
ctp.corr(method = "pearson") #method par defaut pearson de toute facon
```

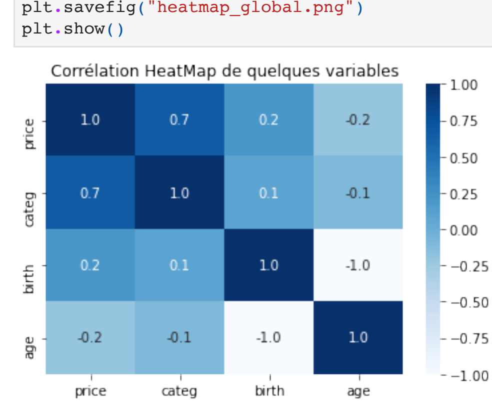
Out[5]:

	price	categ	birth	age
price	1.000000	0.668992	0.209673	-0.209673
categ	0.668992	1.000000	0.094140	-0.094140
birth	0.209673	0.094140	1.000000	-1.000000
age	-0.209673	-0.094140	-1.000000	1.000000

Les resultats vairent entre 1(corr max positif) et -1 (corr max inversée). Il semble une bonne corr positif entre categorie et prix. Nous avons deja vu dans la partie 2 que les prix de la categ 0 etaient les moins elevés et ceux dela categ 2 les plus elevés. Nous pouvons voir un certain degré de correlation inverse entre age et prix, donc les plus jeunes qui depenses plus. La corr serais plus marquée si on regardait par exemple uniquement jusqu'a 40/50 ans. Les moins 30 ans achetent bcp de produit allant jusqu'a 250 $alorsquelespersonnesayantplusde30ansrestententresgrandeparieendessousde100$.

In [6]:
```
#help(sa.corr())
```

In [7]:
```
#Nous pouvons rendre le tableu de corr plus visible avec une heatmap, ajoutant des couleurs qui indiquent
#le degrée de corr.
#Méthode .corr() avec par défaut la formule de la corrélation linéaire de Pearson
sns.heatmap(ctp.corr(), annot=True, fmt=".1f", cmap='Blues')#(annot=marquage du %, fmt=nr apres virgule, cmap=colormap)
plt.title('Corrélation HeatMap de quelques variables')
plt.savefig("heatmap_global.png")
plt.show()
```



In [8]:
```
#help(sns.heatmap)
```

In [9]:
```
#Est-ce qu'il y a une corr entre sex et la categorie des produits, une preference des hommes ou femmes
#pour une categorie
#Création du tableau de contingence "Matrice des valeurs observées"
#Variables 'sex' et 'categ' du dataframe df suivant les instructions du cours
#calcul de chi2, comparaison des valeurs attendus au valeurs obtenus
#Chi2 peut etre appliquer pour deux variables qualitatifs ou deux variables quantitatifs.
#Ici nous avons deux variables categoriques: sex et categorie des produits

X = 'sex'
Y = 'categ'

#Calcul du tableau de contigence par la méthode .pivot_table() et comptages d'achat de chaque categorie
c = ctp[["sex", "categ"]].pivot_table(index=X, columns=Y,  aggfunc=len) # pas compris aggfunc???
tx = ctp[X].value_counts()#value_counts compte le nombre totale de la vaiable choisit (sex, f/m) pour chaque categorie
ty = ctp[Y].value_counts()

#tx est le nombre totale de femmes ou hommes pour les 3 categories
#ty est le nombre totale de femmes + hommes par categorie
# ces chiffres sont neccesaire pour le calcul des valeurs attendu.

#Création d'une copie du dataframe original
cont = c.copy()
#cont["sum"] = ctp["sex"].value_counts()
#cont["sum_"] = ctp["categ"].value_counts()
cont
```

Out[9]:

categ	0.0	1.0	2.0
sex			
f	94728	54657	7703
m	96043	53412	8689

tableu de contingence

In [10]:
```
#Juste pour info je visualise tx et ty
tx
```

Out[10]:
```
m    158144
f    157088
Name: sex, dtype: int64
```

In [11]:
```
ty
```

Out[11]:
```
0.0    190771
1.0    108069
2.0     16392
Name: categ, dtype: int64
```

In [12]:
```
#help(pd.pivot_table)
```

In [13]:
```
#Création de la "Matrice des valeurs attendues"
#L'occurrence attendue est simplement la fréquence que l'on devrait trouver dans une cellule
#si l'hypothèse nulle était vraie.
tx_df= pd.DataFrame(tx)
ty_df = pd.DataFrame(ty)

tx_df.columns = ["s"] #
ty_df.columns = ["s"]

#Valeurs totales observées
n = len(ctp)

#Produit matriciel. On utilise pd.T pour pivoter une des deux séries.
indep = (tx_df.dot(ty_df.T) / n) #matrices des effectifs theoriques
indep
```

Out[13]:

	0.0	1.0	2.0
m	95705.033195	54215.510913	8223.455893
f	95065.966805	53853.489087	8168.544107

In [14]:
```
#Matrice "écart au carré normalisé de la valeur attendue VS valeur observée"
mesure = (c-indep)**2/indep #calcul de chi2. c matrics des effectifs observés/matrices de contengance
mesure
```

Out[14]:

categ	0.0	1.0	2.0
f	1.201498	11.988634	26.532429
m	1.193475	11.908581	26.355260

In [15]:
```
#Calcul du Chi2
#Tester l'hypothèse nulle consiste à comparer les occurrences observées (celles déjà dans le tableau)
#avec les occurrences attendues.
chi2 = mesure.sum().sum() #chaque sum fait la somme d'une dimension x y
chi2
```

Out[15]:
```
79.17987649877874
```

In [16]:
```
#Calcul du khi2 et de la p-value à partir de la matrice des valeurs observées avec scipy
#Degré de liberté = (nombre de lignes – 1) X (nombre de colonnes – 1)
chi2, pvalue, degrees, expected = st.chi2_contingency(cont)
chi2, degrees, pvalue
```

Out[16]:
```
(79.17987649877874, 2, 6.4018910578098306e-18)
```

Conclusion: Pour dl 2 et un seuil de significativité de 5%, le chi2-theorique donné par la table de chi2 est de 5.99. Le chi2 calculé est de 79.17 > 5.99. On rejette donc H0. Il y a donc une correlation entre le sex et la categorie.

In [17]:
```
#Quelques analyses bivariées
#Agrégation pour sommer les ventes 'price' (produits achetés) en fonction de l'âge des clients
#Création d'une variable 'age_price'
age_price = ctp.groupby('age').sum().reset_index()
age_price = age_price[['age', 'price']].sort_values(by='age', ascending=False)
age_price['price'] = age_price['price'] / 1000 #Valeurs exprimées en K€

age_price.head() #Apperçu des données âges / ventes
```

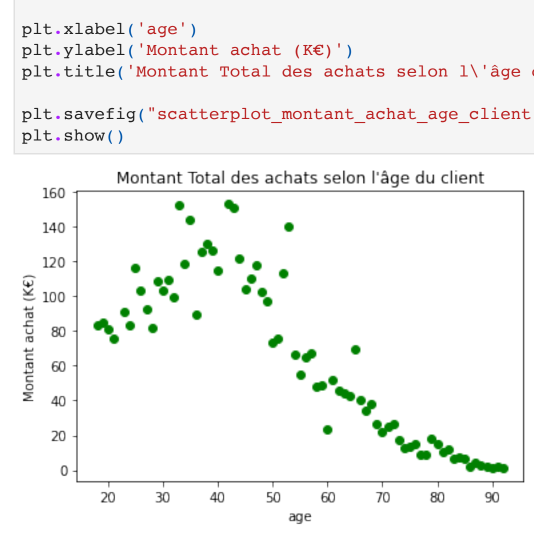
Out[17]:

	age	price
75	92	1.27526
74	91	1.97372
73	90	1.30866
72	89	2.34654
71	88	2.46730

In [18]:
```
#Visualisation avec un scatterplot (âge clients vs montant total des achats)
plt.plot(age_price[age_price.price < 200].age, age_price[age_price.price < 200].price, 'o', color='green')

plt.xlabel('age')
plt.ylabel('Montant achat (K€)')
plt.title('Montant Total des achats selon l\'âge du client')

plt.savefig("scatterplot_montant_achat_age_client.png")
plt.show()
```



In [19]:
```
#Coefficient de corrélation linéaire de Pearson R2
coef_age_price = st.pearsonr(age_price.age, age_price.price)[0]
coef_age_price
```

Out[19]:
```
-0.7747372596963771
```

Le coefficient de pearson indique une correlation lineaire. Il varie entre 0(pas de corr lineaire) et 1(max corr lineaire) valeur absol = 0.77. proche de 1 donc corr age et prix panier moyenne Nous pouvons voir sur le graphe deux et eventuellement trois partie ou on peut tracer ligne. - 18 a 35 ans, corr positif - env 38 a 50 ans corr negatif - 50 a 90 ans corr negatif

In [20]:
```
#help(st.pearsonr)
```

In [21]:
```
#Agrégation des données selon l'âge client
#Le nombre d'achat mensuel est obtenu à partir du comptage des sessions clients par mois
#Hypothèse 1 id_session = 1 transation
customers_freq = ctp.groupby('age').count().reset_index()
customers_freq = customers_freq[['age', 'session_id']]

#Création d'une variable fréquence 'f'
customers_freq['f'] = customers_freq['session_id'] / sum(customers_freq['session_id'])
customers_freq.sort_values(by='age', ascending=False).head(10)
```

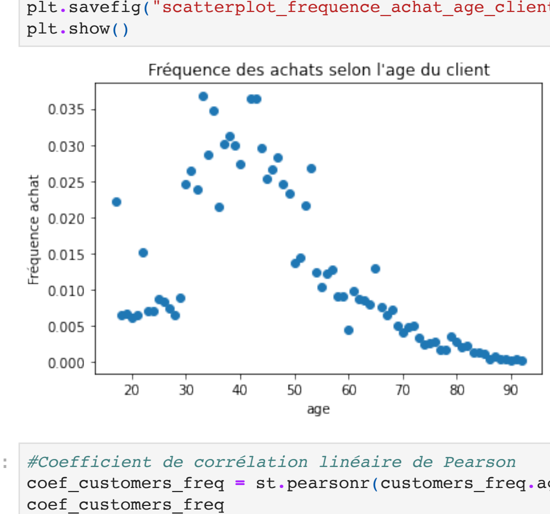
Out[21]:

	age	session_id	f
75	92	84	0.000266
74	91	111	0.000352
73	90	78	0.000247
72	89	141	0.000447
71	88	152	0.000482
70	87	256	0.000812
69	86	132	0.000419
68	85	379	0.001202
67	84	435	0.001380
66	83	416	0.001320

In [22]:
```
#Visualisation avec un scatterplot (âge client vs fréquence d'achat mensuelle)
#customers_freq.plot.scatter(x = 'age', y = 'f', marker = 'o', color='purple')
plt.plot(customers_freq[customers_freq.f < .05].age, customers_freq[customers_freq.f < .05].f, 'o')

plt.xlabel('age')
plt.ylabel('Fréquence achat')
plt.title('Fréquence des achats selon l\'age du client')

plt.savefig("scatterplot_frequence_achat_age_client.png")
plt.show()
```



In [23]:
```
#Coefficient de corrélation linéaire de Pearson
coef_customers_freq = st.pearsonr(customers_freq.age, customers_freq.f)[0]
coef_customers_freq
```

Out[23]:
```
-0.531770132855768
```

valeur absol = 0.53. entre 0 et 1 donc corr leger entre age et frequence d'achat. Mais entre 18 - 30 ans et 50 et 90 ans on pourrait tracer une droite. de 30 a env 53 ans plus tot une forme de cloche qui bruit la corr des autres parties

In [24]:
```
#Panier moyen
#Première agrégation selon l'age client et les sessions en comptage de modalités
customers_shop = ctp.groupby(['age', 'session_id']).count().reset_index()

#Seconde agrégation selon l'age client en moyenne de produits achetés
customers_shop = customers_shop.groupby('age').mean().reset_index()
customers_shop = customers_shop[['age', 'id_prod']]
customers_shop.tail()
```

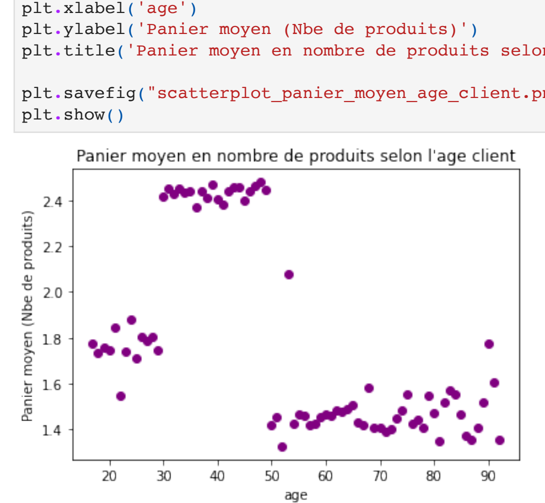
Out[24]:

	age	id_prod
71	88	1.407407
72	89	1.516129
73	90	1.772727
74	91	1.608696
75	92	1.354839

In [25]:
```
#Visualisation avec un scatterplot (âge client vs taille panier moyen)
plt.plot(customers_shop.age, customers_shop.id_prod, 'o', color='purple')

plt.xlabel('age')
plt.ylabel('Panier moyen (Nbe de produits)')
plt.title('Panier moyen en nombre de produits selon l\'age client')

plt.savefig("scatterplot_panier_moyen_age_client.png")
plt.show()
```



In [26]:
```
coef_age_price = st.pearsonr(age_price.age, age_price.price)[0]
coef_age_price
```

Out[26]:
```
-0.7747372596963771
```

valeur abs 0.77 donc forte corr lineaire. On peut clairement distunger 3 parties ou on peut tracer une droite. - jusqu'a 30ans - 30 a 50 ans - 50 a 90 ans

In [27]:
```
#Analyse de la corrélation entre l'âge clients et la catégorie produits
#N est le nombre d'observations, ici représentées par les valeurs transactionnelles par âge et par catégorie
len(ctp.groupby(['age', 'categ']).count().reset_index())
```

Out[27]:
```
227
```

In [28]:
```
#Méthode .groupby() pour agréger les données selon l'âge et la catégorie
age_categ = ctp.groupby(['age', 'categ']).count().reset_index()
age_categ = age_categ[['age', 'categ', 'session_id']]
age_categ.head()
```

Out[28]:

	age	categ	session_id
0	17	0.0	1534
1	17	1.0	2711
2	17	2.0	2725
3	18	0.0	429
4	18	1.0	801

In [29]:
```
#Méthode .cut() pour créer les 9 groupes d'âges, une segmentation des individus 'age'
age_categ['age']= pd.cut(age_categ['age'], 9)
age_categ = age_categ.groupby(['age','categ']).sum().reset_index()
age_categ.head(10)
```

Out[29]:

	age	categ	session_id
0	(16.925, 25.333]	0.0	5445
1	(16.925, 25.333]	1.0	10390
2	(16.925, 25.333]	2.0	11223
3	(25.333, 33.667]	0.0	28851
4	(25.333, 33.667]	1.0	12172
5	(25.333, 33.667]	2.0	3966
6	(33.667, 42.0]	0.0	73939
7	(33.667, 42.0]	1.0	23541
8	(33.667, 42.0]	2.0	262
9	(42.0, 50.333]	0.0	47958

In [30]:
```
#Visualisation rapide avec une BarPlot Seaborn
fig, ax = plt.subplots(figsize=(22, 12))
sns.barplot(x="age", y="session_id",hue="categ", data=age_categ)

ax.set_xlabel('age')
ax.set_ylabel('Nombre de produits vendus')
ax.set_title('Catégories produits selon l\'âge client')

plt.savefig("barplot_categorie_produit_age_client")
plt.show()
```

On peut observer que en dessous de 25 ans les produits des categories 1 + 2 sont predominantes, alors qu'a partir de 25ans les produits de la categorie 0 dominent. A aprtir de 50 ans les produits de la categorie 1 predominent legerement devant les produits de la categorie 0. Il y a donc 3 groupes.

In [32]:
```
#Analayse anova categorie versus prix. Il permet d'analyser la relation entre une
#variable qualitatif et quantitatif. Pour simplicité de calcul je regarde la categorie (modalité 0, 1 et 2)
#en fonction de l'age. Malheuresement pour les autres variables categoriques l'ordinateur
#ou plus tot la connexion a rompu systematiquement.
fit = statsmodels.formula.api.ols( "categ ~ age", data = ctp).fit()
table = statsmodels.api.stats.anova_lm(fit)
table
```

Out[32]:

	df	sum_sq	mean_sq	F	PR(>F)
age	1.0	981.063600	981.063600	2818.647362	0.0
Residual	315230.0	109719.535245	0.348062	NaN	NaN

Conclusion: calcul df pour anova: cat 3: p-1(3-1=2),v1; residual.: n-p(ligne -3)(336816-3), v2. ensuite on compare au tableau de fischer: si la valeur calculé est plus grand que la valeur du tableau de fischer on peut rejetter H0 et dire que les variables sont correlés. F=2818,6 >> 2.99 (tableau de fischer pour v1=2; v2>120) >> H0 peut rejetté, les variables age et categorie sont fortement correlés.

In [34]:
```
#ctp["age"].unique
```

In [ ]: