



PROJET 5 :

# SEGMENTATION DES CLIENTS DU SITE E- COMMERCE :

# Mission et objectifs

- **Mission**

OLIST, site d' e-commerce souhaite fournir à ses équipes marketing une segmentation de ses clients utilisable au quotidien pour leurs campagnes de communication

- **Objectifs**

- Comprendre les différents types d'utilisateurs grâce à leur comportement et à leurs données personnelles
- Fournir à l' équipe marketing une description actionnable de notre segmentation et de sa logique sous-jacente pour une utilisation optimale
- Réaliser une proposition de contrat de maintenance basée sur une analyse de la stabilité des segments au cours du temps

# Sommaire

- I : Exploration et transformation des données
  - Chargement et aperçu des données
  - Détection des valeurs manquantes
  - Nettoyage des données
  - Origines Clients et Vendeurs
  - Comportement des clients
  - Produits
  - Commandes
- II : Élaboration des modèles de clustering
  - Normalisation des données
  - K-Means
  - DBSCAN
  - Clustering hiérarchique agglomératif
- III : Maintenance
  - Chargement et aperçu des données

# *Analyse Exploratoire*

# Chargement et aperçu de données

- Chargement, formats et structures des données :

```
Clients = pd.read_csv('data/olist_customers_dataset.csv')
Geolocation = pd.read_csv('data/olist_geolocation_dataset.csv')
Items_commandes = pd.read_csv('data/olist_order_items_dataset.csv')
Paiements_commandes = pd.read_csv('data/olist_order_payments_dataset.csv')
Avis_commande = pd.read_csv('data/olist_order_reviews_dataset.csv')
```

```
Commandes = pd.read_csv('data/olist_orders_dataset.csv')
Produits = pd.read_csv('data/olist_products_dataset.csv')
Vendeurs = pd.read_csv('data/olist_sellers_dataset.csv')
Trad_cat_produits = pd.read_csv('data/product_category_name_translation.csv')
```

Clients	
Nbre de lignes	Nbre de variables
99441	5

Types de variables			
Objet	Float	Int	Bool
4	0	1	0

Colonnes/Lignes dupliquées	
Colonnes dupliquées	Lignes dupliquées
0	0

Geolocation	
Nbre de lignes	Nbre de variables
1000163	5

Types de variables			
Objet	Float	Int	Bool
2	2	1	0

Colonnes/Lignes dupliquées	
Colonnes dupliquées	Lignes dupliquées
0	261831

Items_commandes	
Nbre de lignes	Nbre de variables
112650	7

Types de variables			
Objet	Float	Int	Bool
4	2	1	0

Colonnes/Lignes dupliquées	
Colonnes dupliquées	Lignes dupliquées
0	0

Paiements_commandes	
Nbre de lignes	Nbre de variables
103886	5

Types de variables			
Objet	Float	Int	Bool
2	1	2	0

Colonnes/Lignes dupliquées	
Colonnes dupliquées	Lignes dupliquées
0	0

Avis_commande	
Nbre de lignes	Nbre de variables
99224	7

Types de variables			
Objet	Float	Int	Bool
6	0	1	0

Colonnes/Lignes dupliquées	
Colonnes dupliquées	Lignes dupliquées
0	0

Commandes	
Nbre de lignes	Nbre de variables
99441	8

Types de variables			
Objet	Float	Int	Bool
8	0	0	0

Colonnes/Lignes dupliquées	
Colonnes dupliquées	Lignes dupliquées
0	0

Produits	
Nbre de lignes	Nbre de variables
32951	9

Types de variables			
Objet	Float	Int	Bool
2	7	0	0

Colonnes/Lignes dupliquées	
Colonnes dupliquées	Lignes dupliquées
0	0

Vendeurs	
Nbre de lignes	Nbre de variables
3095	4

Types de variables			
Objet	Float	Int	Bool
3	0	1	0

Colonnes/Lignes dupliquées	
Colonnes dupliquées	Lignes dupliquées
0	0

Trad_cat_produits	
Nbre de lignes	Nbre de variables
71	2

Types de variables			
Objet	Float	Int	Bool
2	0	0	0

Colonnes/Lignes dupliquées	
Colonnes dupliquées	Lignes dupliquées
0	0

- Détection des valeurs manquantes

**Clients**

	count	percent
customer_id	0	0.0
customer_unique_id	0	0.0
customer_zip_code_prefix	0	0.0
customer_city	0	0.0
customer_state	0	0.0

**Geolocation**

	count	percent
geolocation_zip_code_prefix	0	0.0
geolocation_lat	0	0.0
geolocation_lng	0	0.0
geolocation_city	0	0.0
geolocation_state	0	0.0

**Items\_commandes**

	count	percent
order_id	0	0.0
order_item_id	0	0.0
product_id	0	0.0
seller_id	0	0.0
shipping_limit_date	0	0.0
price	0	0.0
freight_value	0	0.0

**Paiements\_commandes**

	count	percent
order_id	0	0.0
payment_sequential	0	0.0
payment_type	0	0.0
payment_installments	0	0.0
payment_value	0	0.0

**Avis\_commande**

	count	percent
review_id	0	0.000000
order_id	0	0.000000
review_score	0	0.000000
review_creation_date	0	0.000000
review_answer_timestamp	0	0.000000
review_comment_message	58247	58.702532
review_comment_title	87656	88.341530

**Commandes**

	count	percent
order_id	0	0.000000
customer_id	0	0.000000
order_status	0	0.000000
order_purchase_timestamp	0	0.000000
order_estimated_delivery_date	0	0.000000
order_approved_at	160	0.160899
order_delivered_carrier_date	1783	1.793023
order_delivered_customer_date	2965	2.981668

- Détection des valeurs manquantes (suite)

**Produits**

	count	percent
<b>product_id</b>	0	0.000000
<b>product_weight_g</b>	2	0.006070
<b>product_length_cm</b>	2	0.006070
<b>product_height_cm</b>	2	0.006070
<b>product_width_cm</b>	2	0.006070
<b>product_category_name</b>	610	1.851234
<b>product_name_lenght</b>	610	1.851234
<b>product_description_lenght</b>	610	1.851234
<b>product_photos_qty</b>	610	1.851234

**Vendeurs**

	count	percent
<b>seller_id</b>	0	0.0
<b>seller_zip_code_prefix</b>	0	0.0
<b>seller_city</b>	0	0.0
<b>seller_state</b>	0	0.0

**Trad\_cat\_produits**

	count	percent
<b>product_category_name</b>	0	0.0
<b>product_category_name_english</b>	0	0.0

On constate que les DataFrames Avis\_commande, Commandes et Produits possèdent des valeurs manquantes mais seule Avis\_commande comporte une variable dépassant les 88% de nan

# Nettoyage des données

- Traduction des noms des catégories en anglais

Avant

```
0      perfumaria
1          artes
2    esporte_lazer
3        bebes
4  utilidades_domesticas
5 instrumentos_musicais
6       cool_stuff
7 moveis_decoracao
8 eletrodomesticos
9     brinquedos
Name: product_category_name, dtype: object
```

Après

```
0      perfumery
1          art
2 sports_leisure
3        baby
4   housewares
5 musical_instruments
6       cool_stuff
7 furniture_decor
8 home_appliances
9        toys
Name: product_category_name, dtype: object
```

- Transformer les variables contenant des valeurs chronologiques en type DateTime

Avant

```
order_purchase_timestamp      object
order_approved_at            object
order_delivered_carrier_date object
order_delivered_customer_date object
order_estimated_delivery_date object
dtype: object
```

Après

```
order_purchase_timestamp      datetime64[ns]
order_approved_at            datetime64[ns]
order_delivered_carrier_date datetime64[ns]
order_delivered_customer_date datetime64[ns]
order_estimated_delivery_date datetime64[ns]
dtype: object
```

# Merge de toutes nos DataFrames

```
1 df_1 = pd.merge(Clients,Commandes,on="customer_id")
2 df_2 = pd.merge(df_1,Items_commandes,on="order_id")
3 df_3 = pd.merge(df_2,Produits,on="product_id")
4 df_4 = pd.merge(df_3,Avis_commande,on="order_id")
5 df_5 = pd.merge(df_4,Paiements_commandes,on="order_id")
6 df_final = pd.merge(df_5,Vendeurs,on="seller_id")
```

```
1 df_final.shape
```

(114844, 39)

```
1 # supprimer les lignes dupliquées en se basant sur l'id unique des commandes
2 df_final = df_final.drop_duplicates(subset="customer_id")
```

```
1 df_final.shape
```

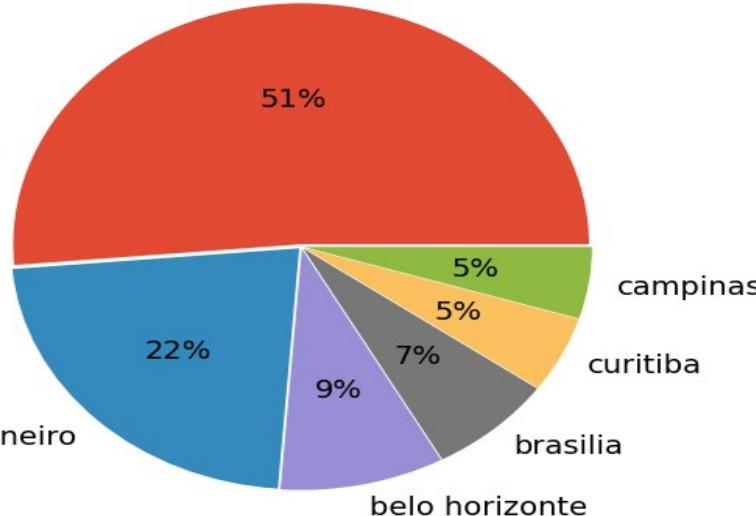
(95817, 39)

# Origines Clients et Vendeurs

- Clients

Origines des clients

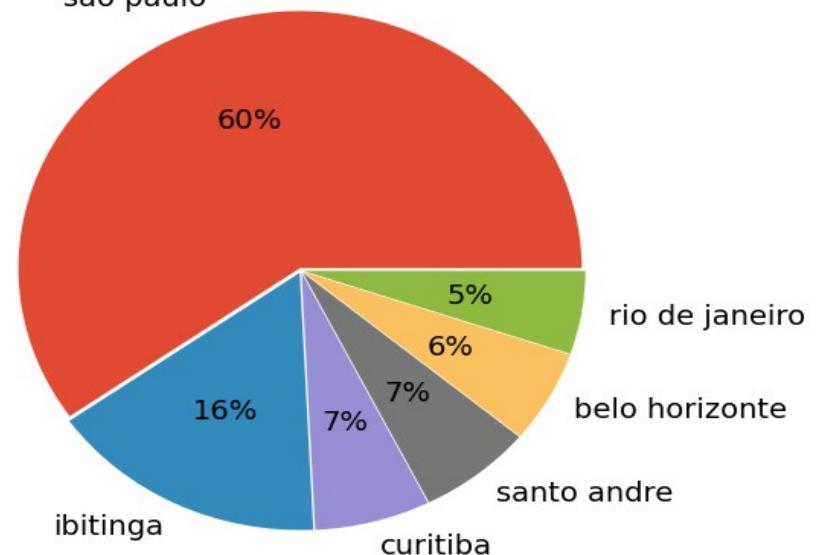
sao paulo



- Vendeurs

Origines des vendeurs

sao paulo

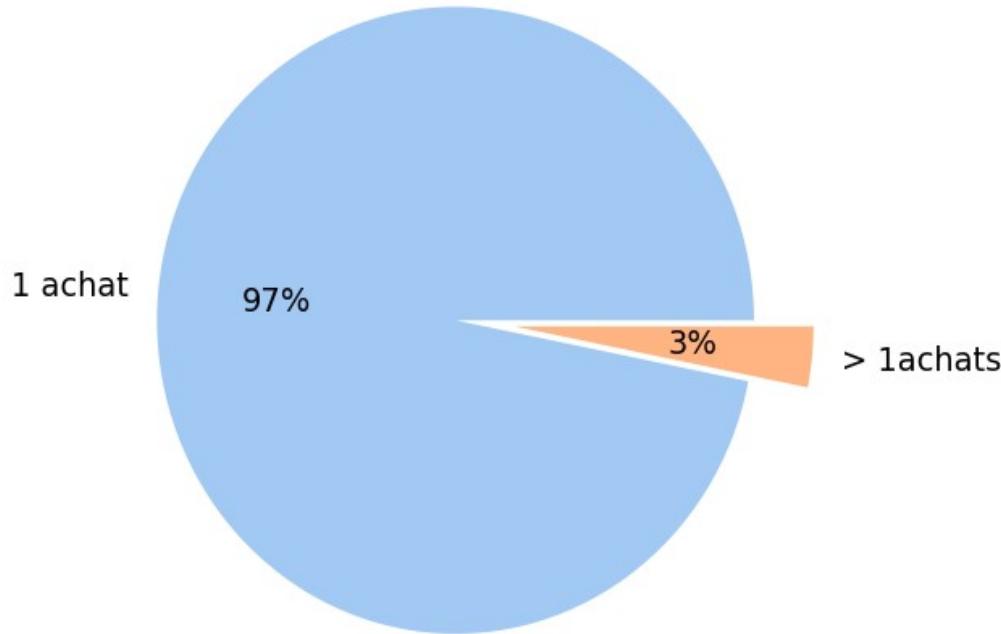


La majorité des clients se situent à Sao Paulo et Rio de Janeiro

La plus part des vendeurs se situent à Sao Paulo

- Comportement des clients

- Proportion des clients par rapport aux nombres d'achats

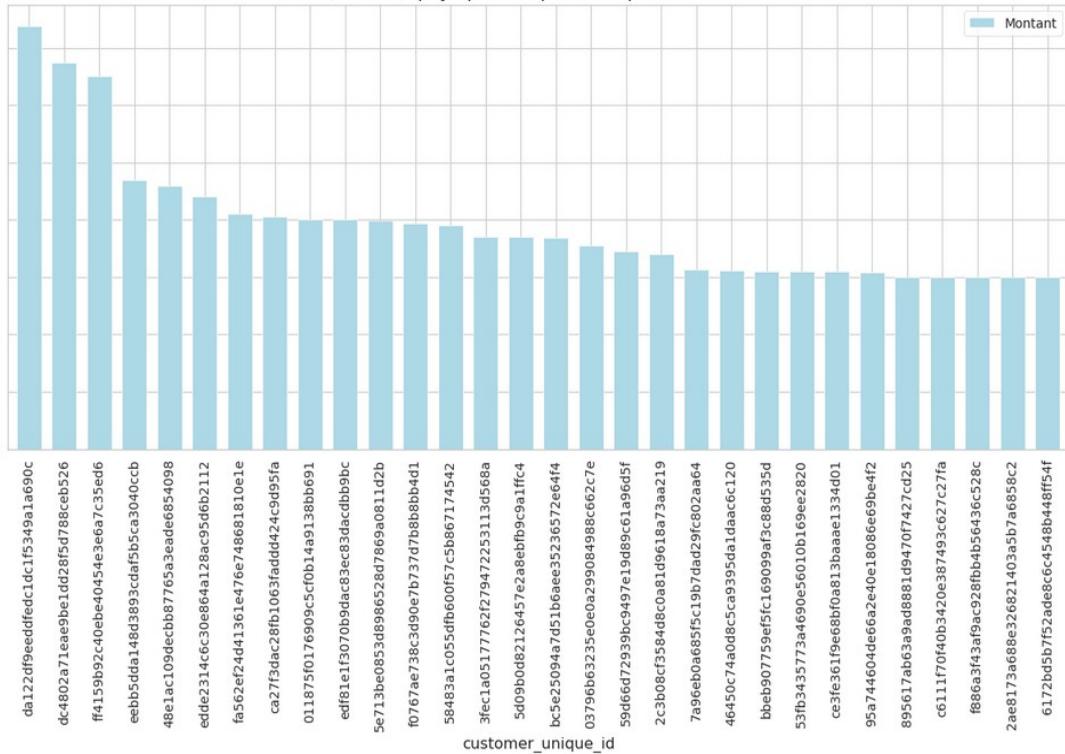


On constate que seulement 3% des clients qui ont procédé à plus d'un achat.

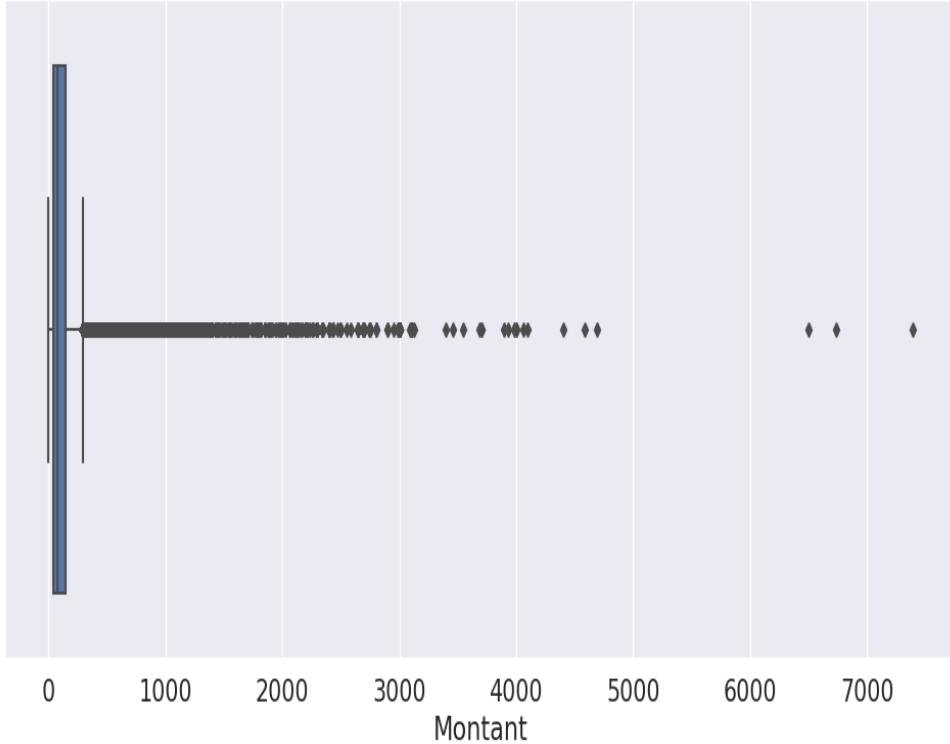
# • Comportement des clients

## • Montant total d'achats par clients

Le montant total(en Réal) payé par chaque client pour l'ensemble de ses achats



boxplot des montants payés par les clients sur l'ensemble des achats

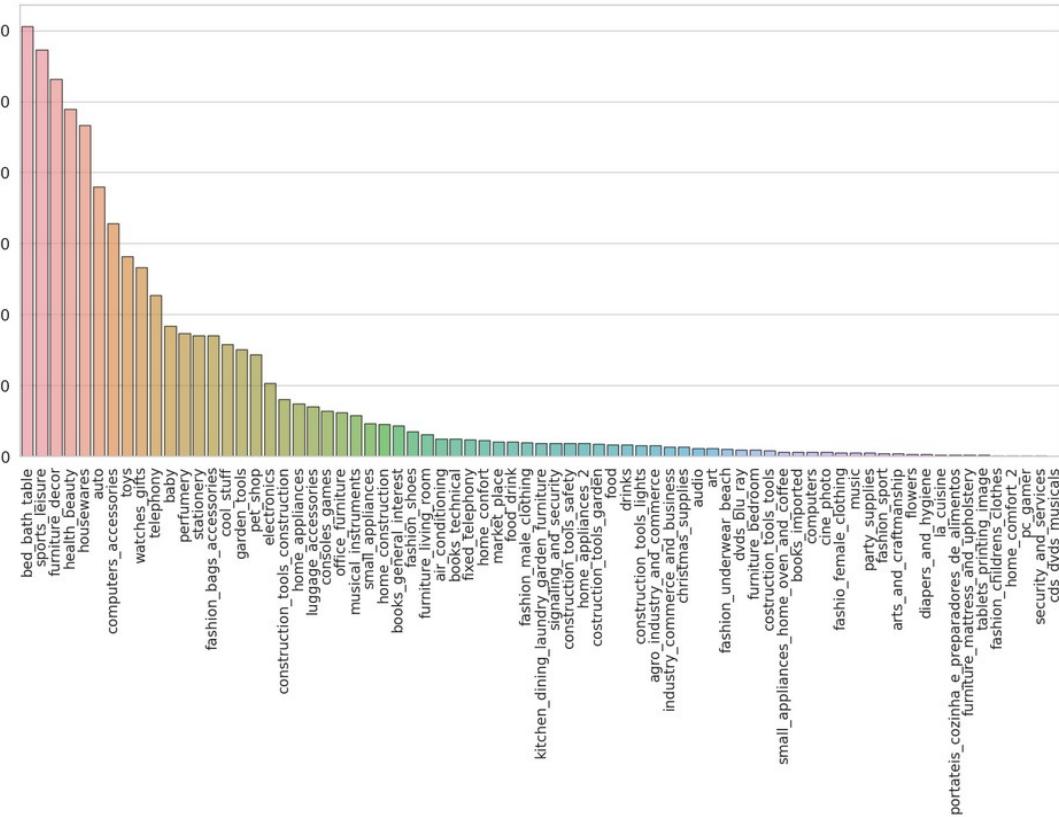


Seulement 6 clients ont fait un total d'achats supérieur à 6000 réal (1000 euros)

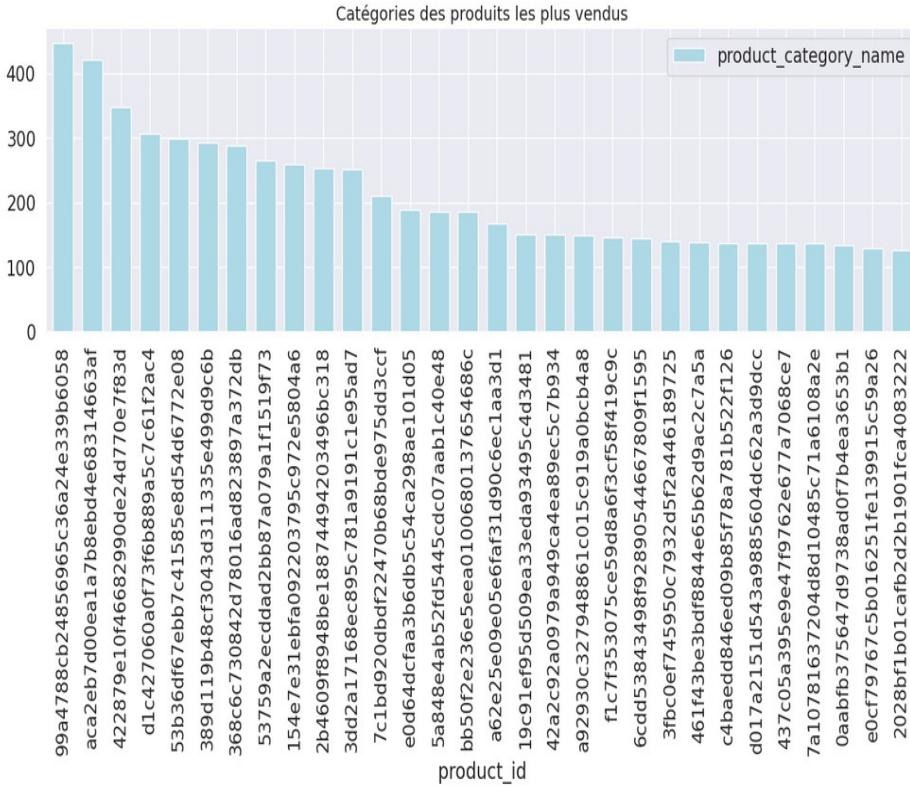
La plus part des clients ont fait des petites dépenses

# Produits

- Catégories des produits:



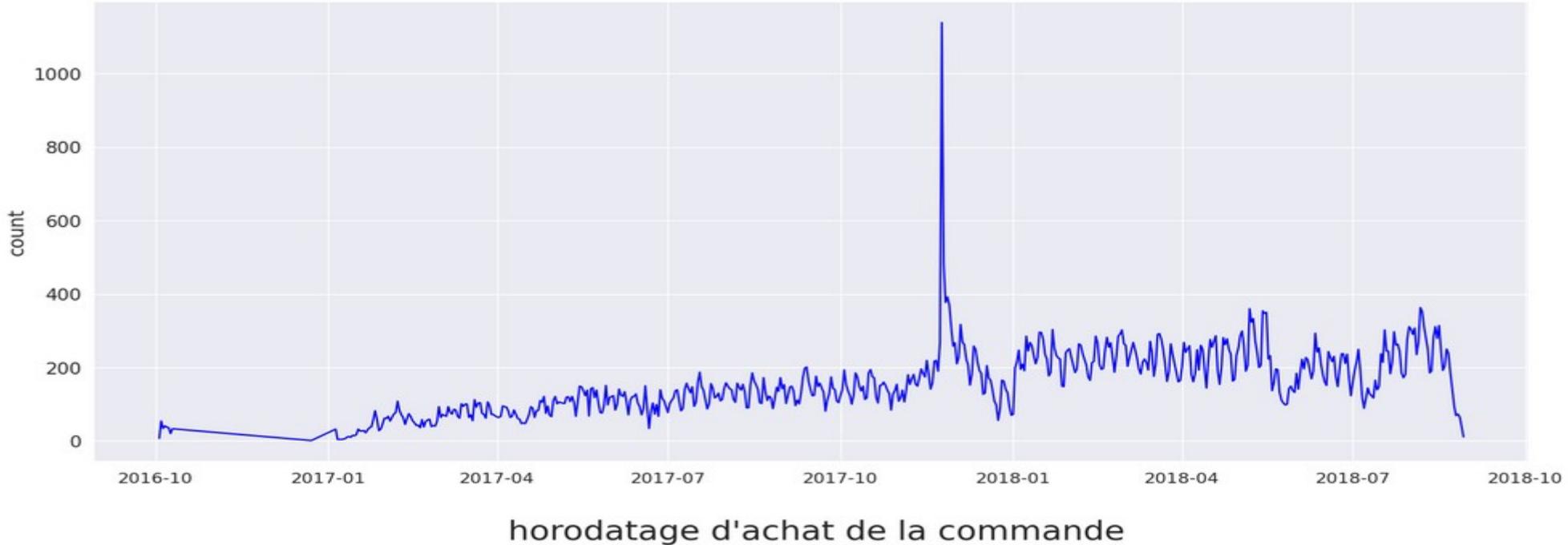
- Catégories des produits les plus vendus:



# Commandes

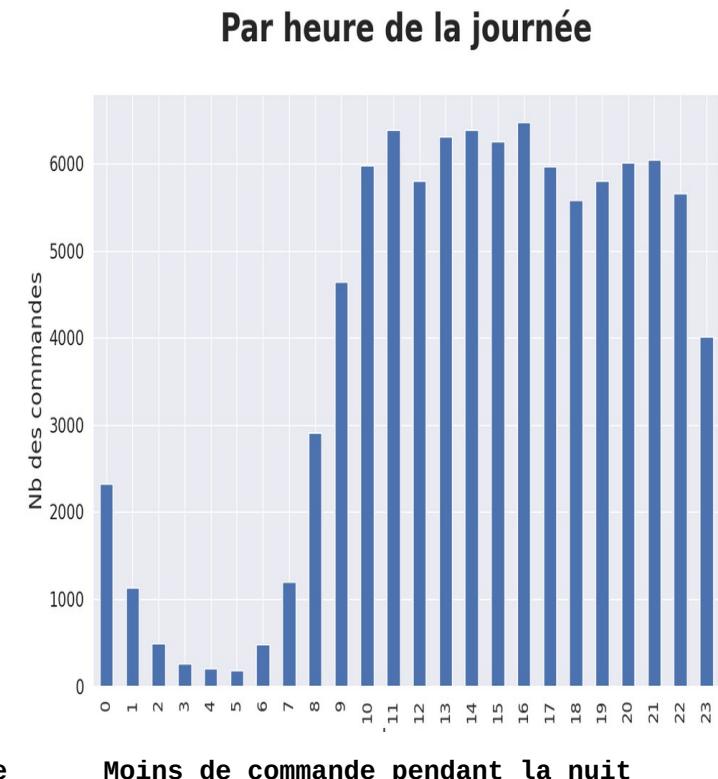
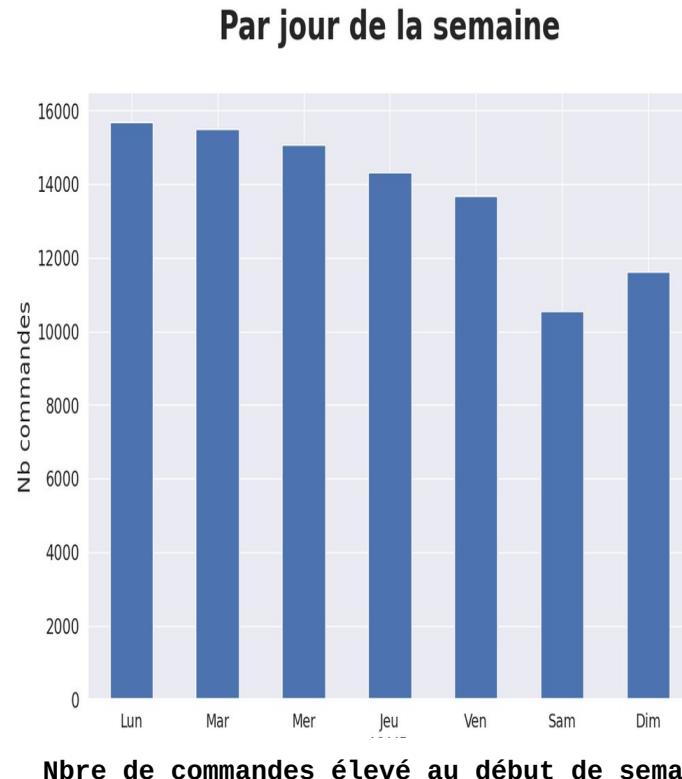
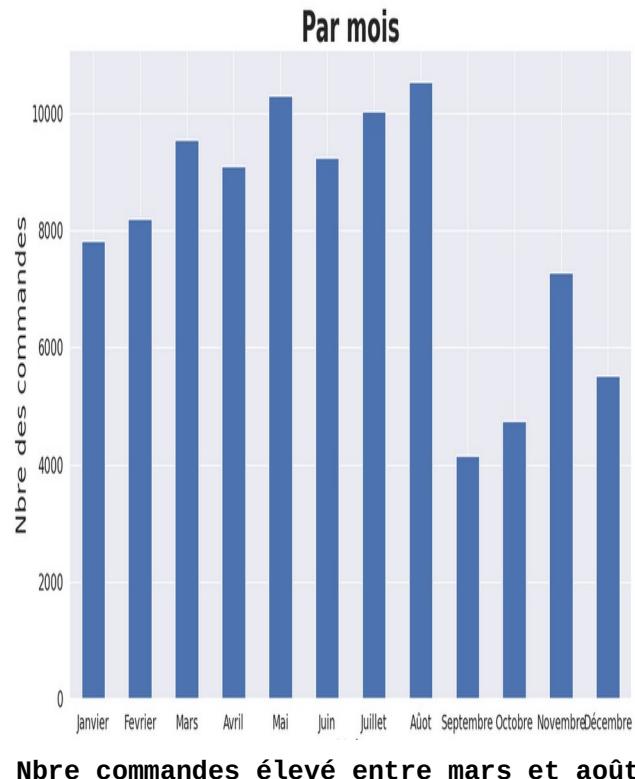
- Évolution du nombre de commandes journalière :

## **Evolution du nombre des commandes journalières**



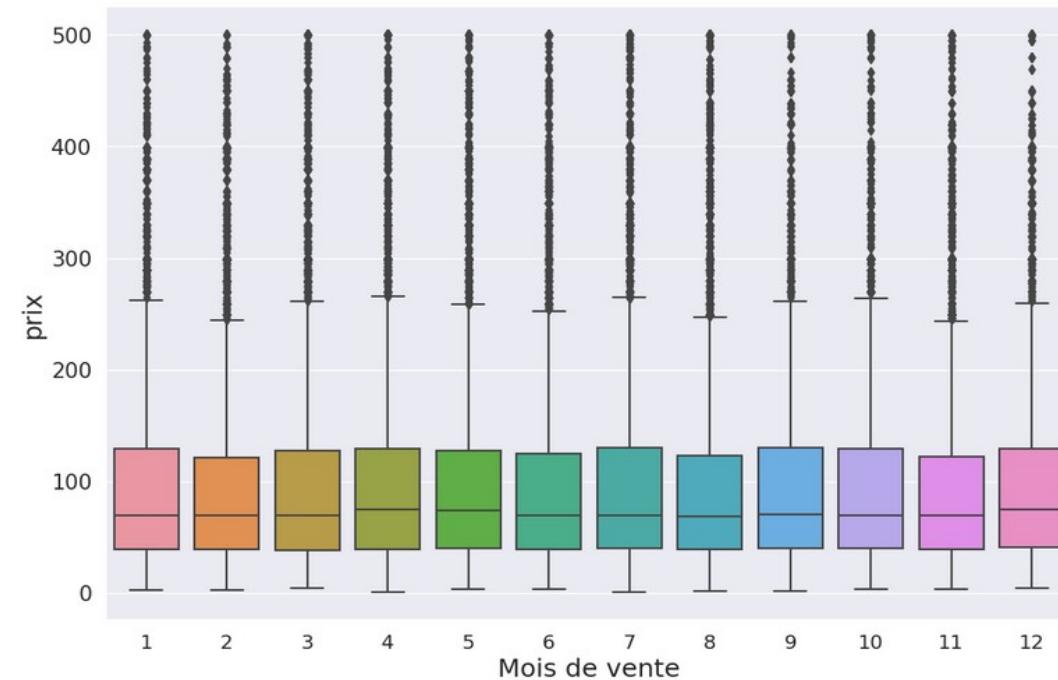
# Commandes

- Répartition des commandes :

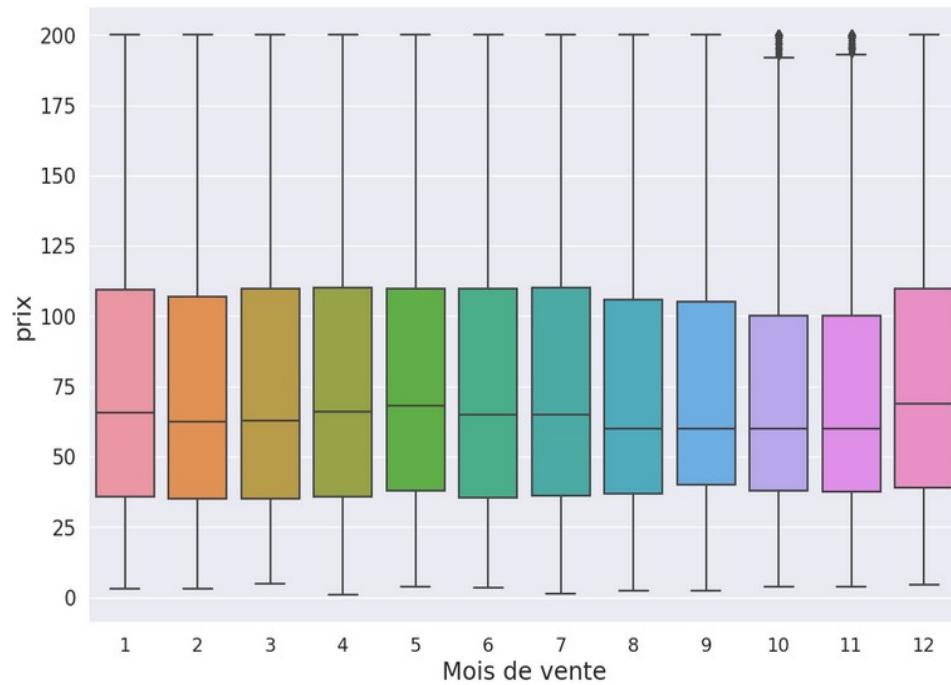


# Chiffre d'affaire par mois

- Distribution du chiffre d'affaire sur les mois de l'année :



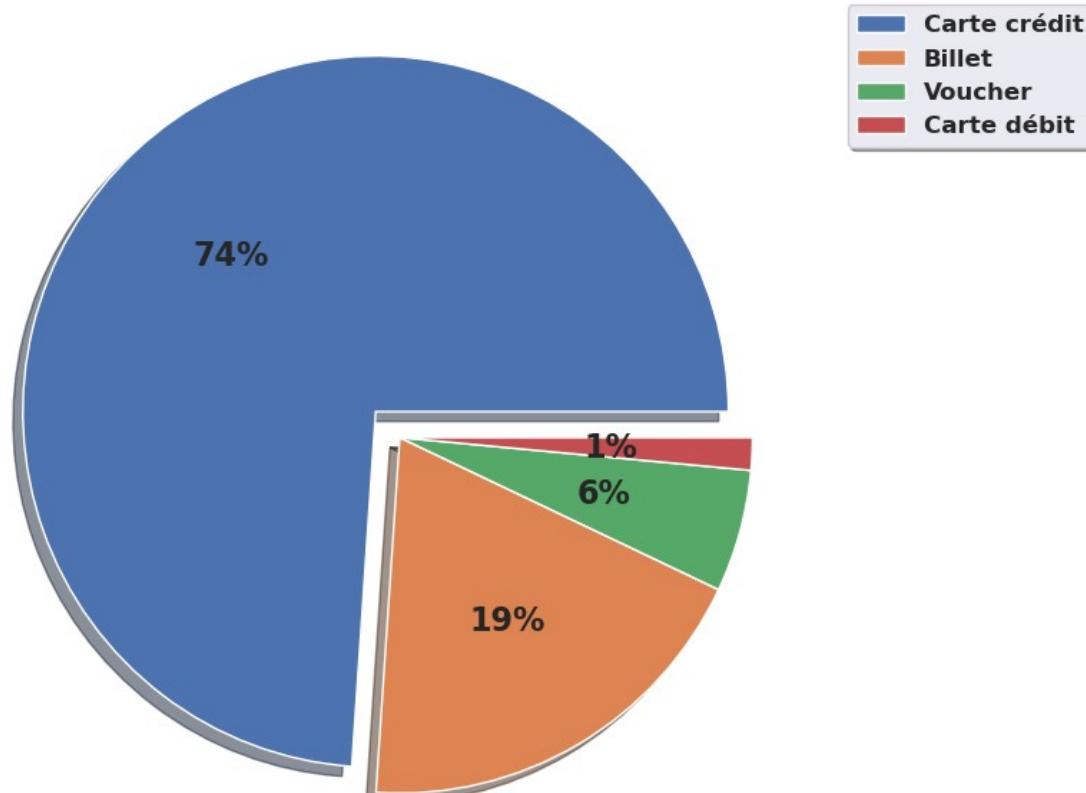
items.price < 500



items.price < 200

# Les moyens de payement

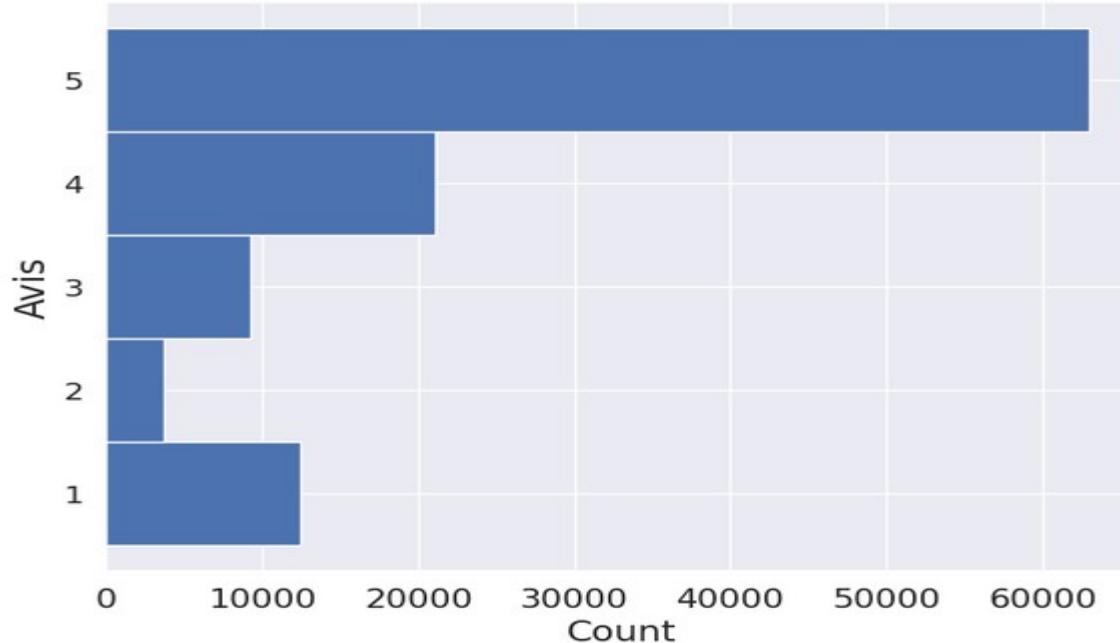
## Les moyens de paiement utilisés



La plus part des paiement sont effectués par carte de crédit

# Avis des clients

- Notes attribuées aux commandes par les clients : [1 à 5]



Les clients, dans leur majorité (> 60 000) donnent des avis favorables aux commandes

# Segmentation RFM

La segmentation RFM permet de cibler les offres, d'établir des segments basés sur la valeur des clients et de prévenir l'attrition en identifiant des segments à risque.

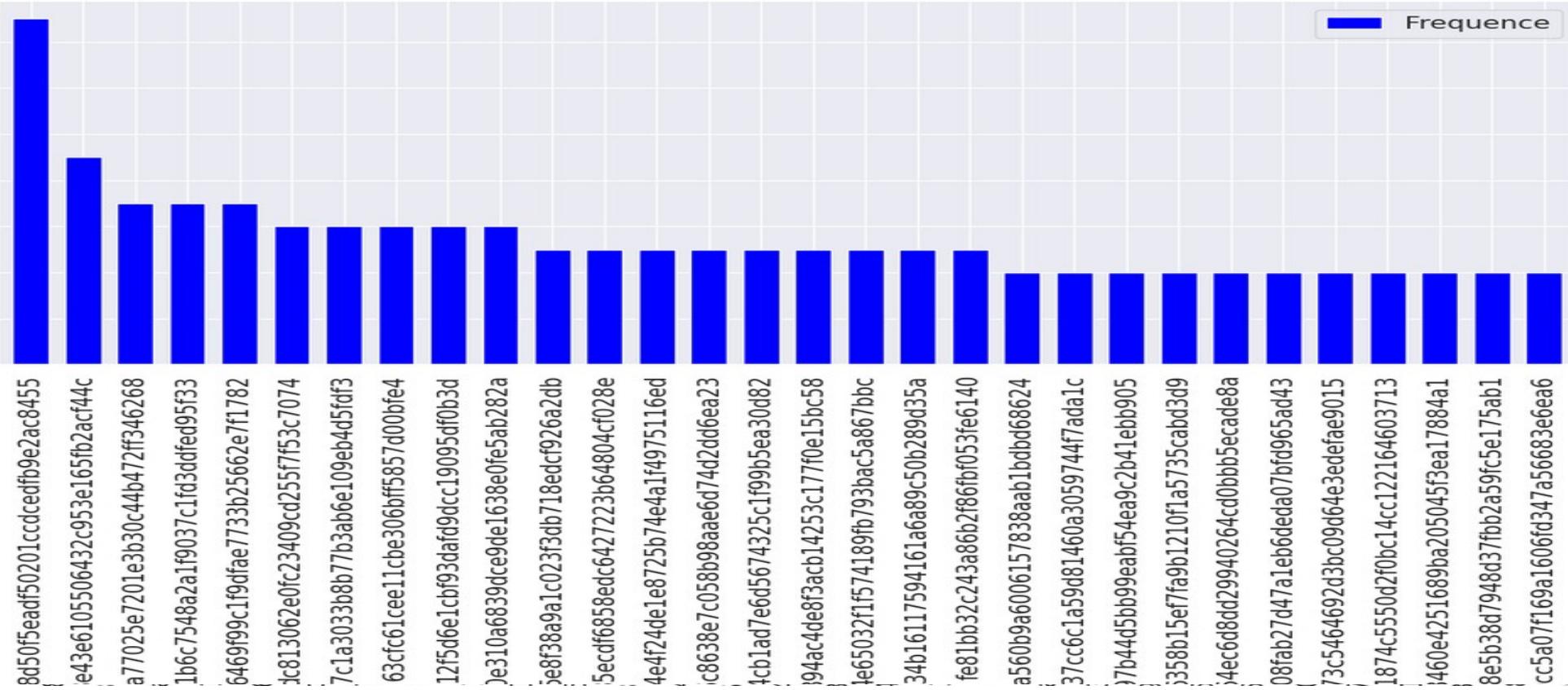
La segmentation RFM prend en compte la Récence (date de la dernière commande), la Fréquence des commandes et le Montant (de la dernière commande ou sur une période donnée) pour établir des segments de clients homogènes.

À cet effet, on va créer trois nouvelles variables : Montant, Fréquence et Récence.

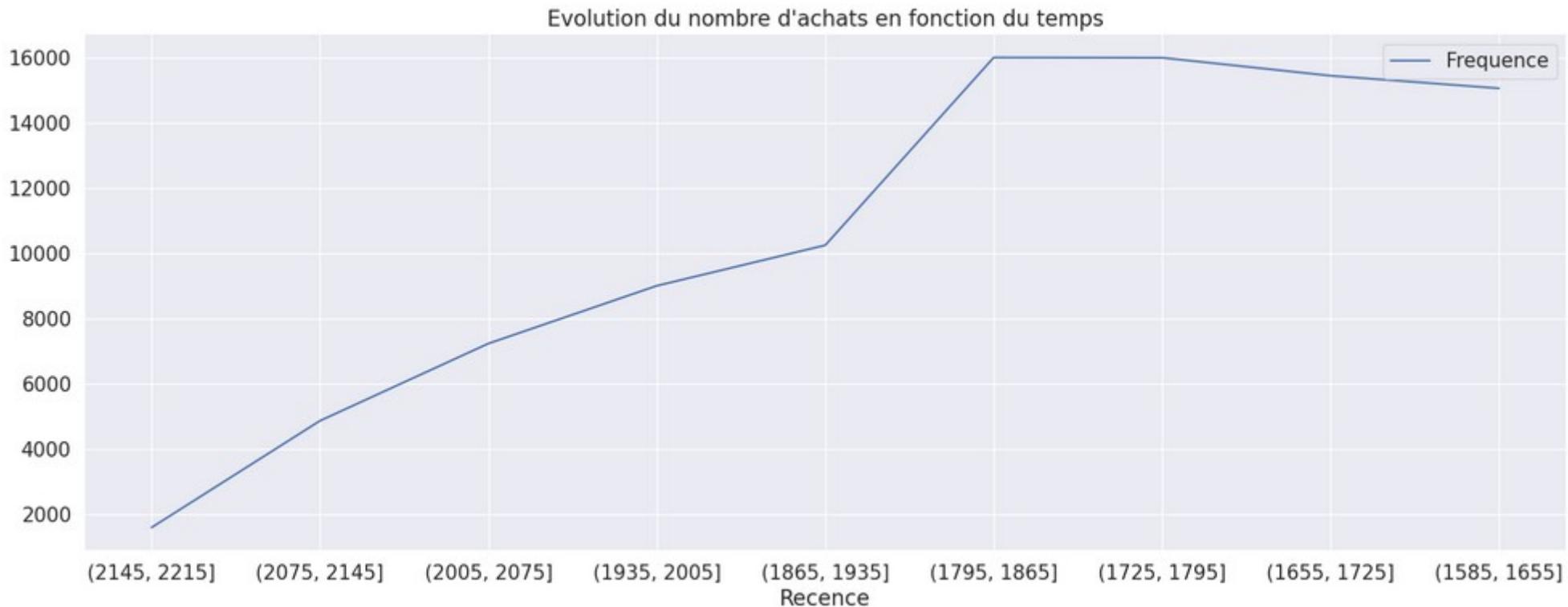
En outre de ces variables nouvellement créées, on créera une nouvelle variable qui prend en compte l'avis des clients : Avis.

# Fréquence

Fréquence d'achats par client



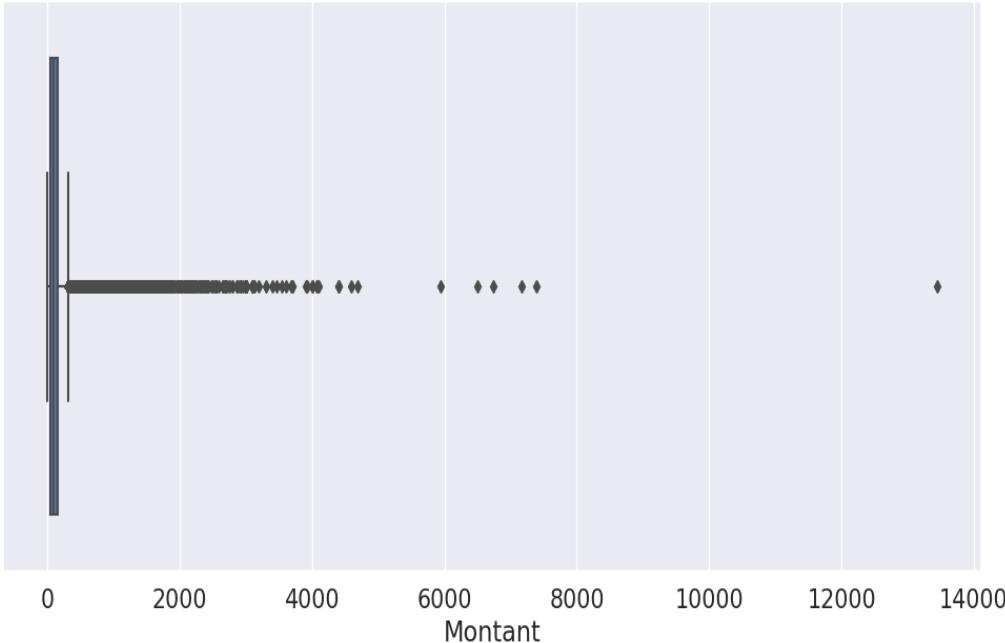
# Nombre d'achats total en fonction du temps



# Segmentation RFM

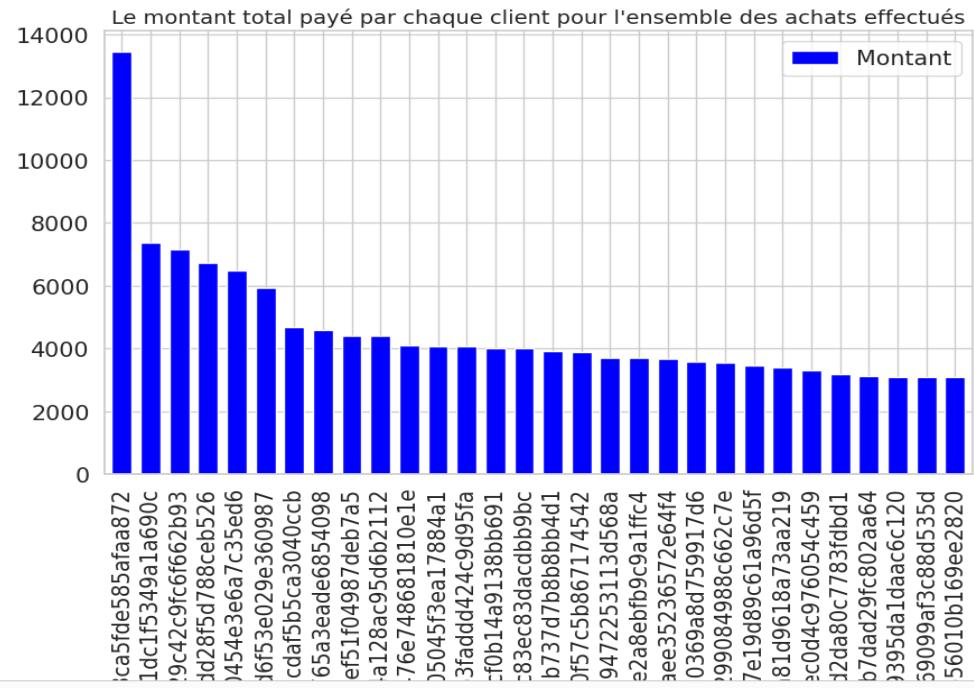
- Montant :

boxplot des montants payés par les clients sur l'ensemble des achats



La plus des clients dépensent des petites sommes en terme d'achats

Les 30 clients qui ont le plus dépensé en terme d'achat



# Segmentation RFM

- Récense

```
1 print("    la plus vieille date de commande : {}".format(data_global_final["order_purchase_timestamp"].min()))
2 print("    la plus récente date de commande : {}".format(data_global_final["order_purchase_timestamp"].max()))

la plus vieille date de commande : 2016-10-03 09:44:50
la plus récente date de commande : 2018-08-29 15:00:37
```

## Construction de la variable Récence

```
1 today = pd.to_datetime("2023-01-11")
2 dt = date.copy()
3 dt["Récence"] = (today - date["order_approved_at"]).dt.days

1 dt = dt.groupby("customer_unique_id").agg("min").reset_index()

1 cluster_table_final = pd.merge(cluster_table,dt,on="customer_unique_id")
```

# Partie II : Modèles de clustering

- **K-Means :**

RFM / RFM + Avis:

- Silhouette
- Calinski-harabasz
- Inertie/Distortion
- Analyse multivariées – Boxplots
- Visualisation 2D des clusters

- **DBSCAN :**

- Tuning des hyper-paramètres
- Visualisation 2D des clusters formés par DBSCAN
- Nombre de clients dans les clusters
- Évaluation du bruit

- **Clustering hiérarchique agglomératif :**

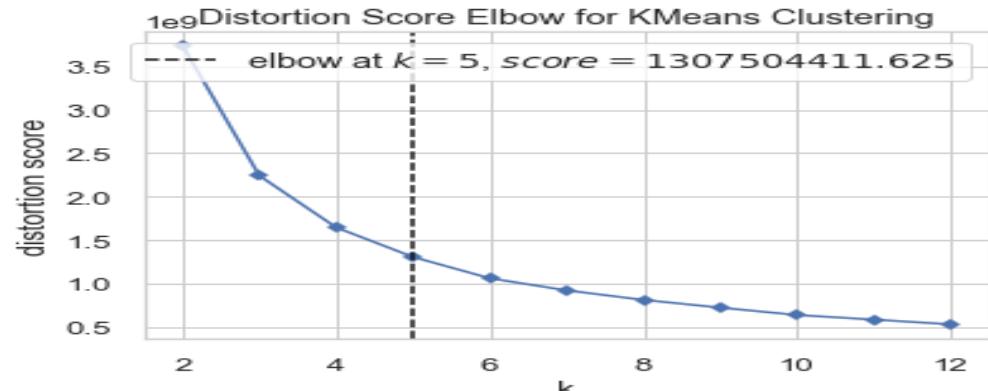
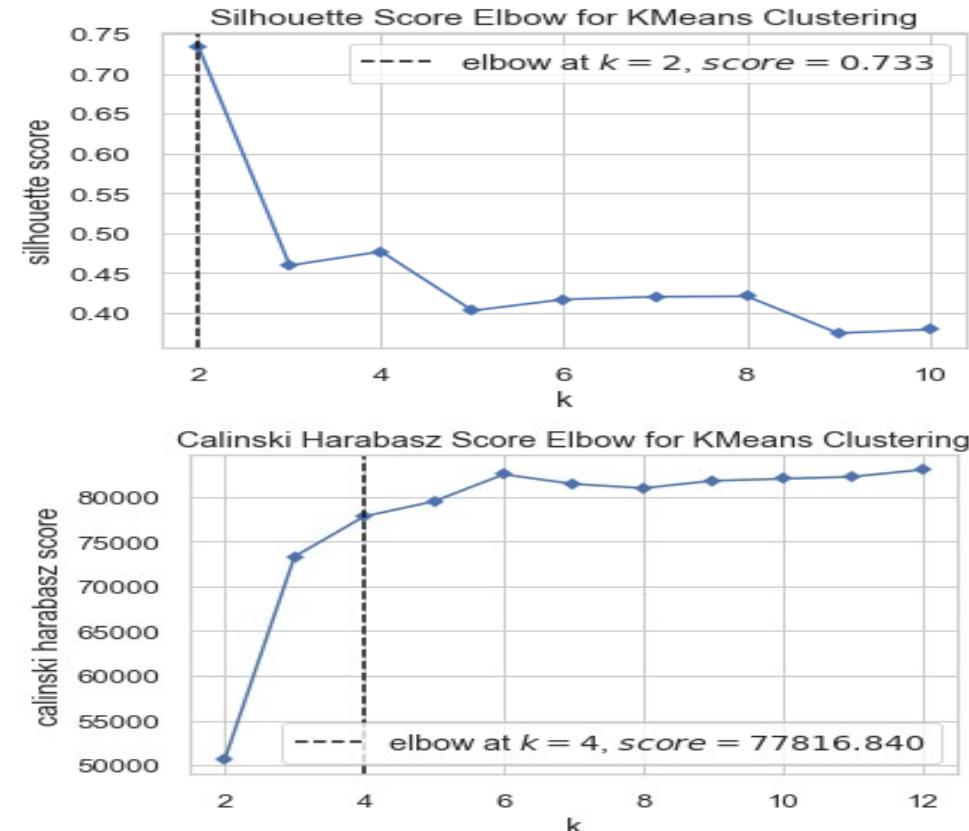
- Visualisation 2D des clusters formés par les algorithmes hiérarchiques
- Nombre de clients dans les clusters avec Ward et complete linkage

## Partie II : Modèles de clustering

- **K-Means :**
- L'algorithme KMeans regroupe les données en essayant de séparer les échantillons en n groupes de variance égale, en minimisant le critère inertie ou somme des carrés intra-cluster. Cet algorithme nécessite que le nombre de clusters k soit spécifié.
- **Démarche :**
  - 1) Faire varier k entre 2 et 10
  - 2) Calculer les performances des clustering (Silhouette, Calinski-Harabasz et distortion)
  - 3) Faire une visualisation des différents clusters obtenus
  - 4) Mesure et analyse orienté métier
  - 5) Choix du meilleur nombre de clusters k

# Partie II : Modèles de clustering

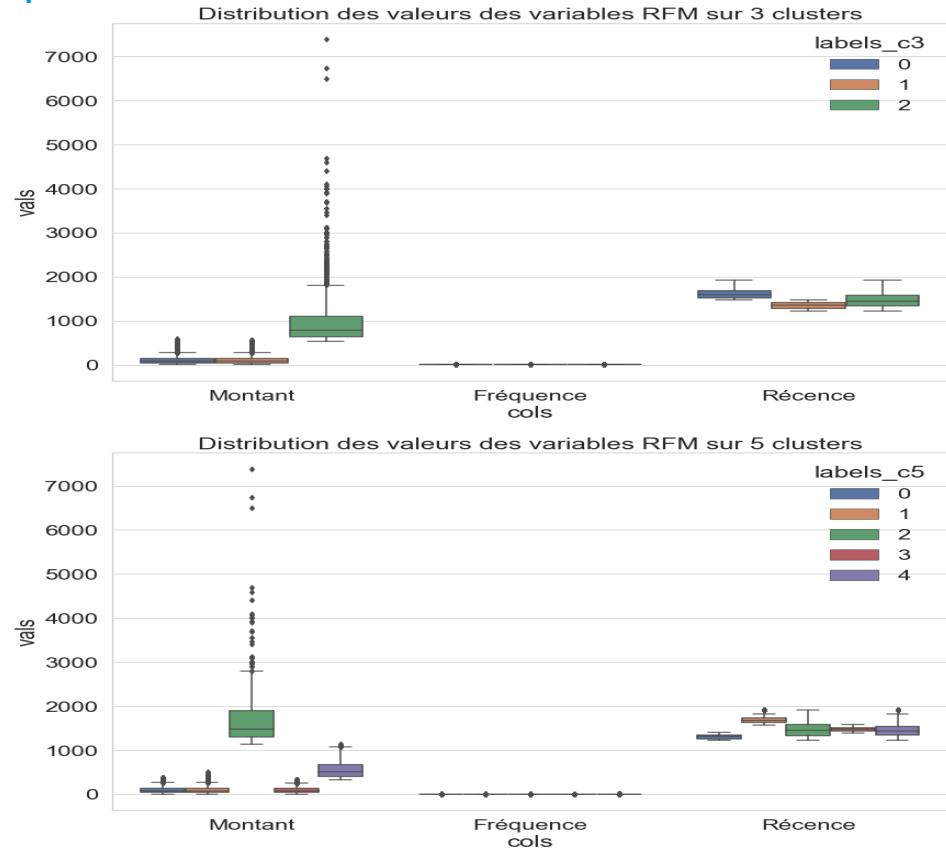
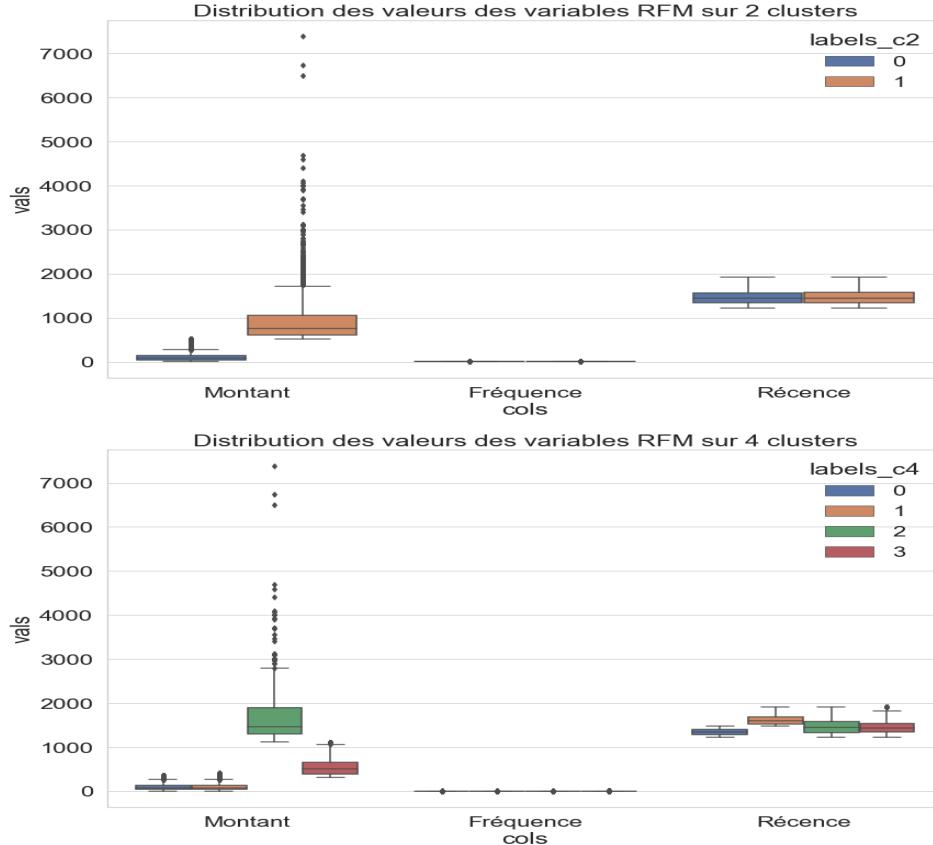
- K-Means : RFM



- Le coefficient de silhouette est la différence entre la distance moyenne avec les points du même groupe que lui et la distance moyenne avec les points des autres groupes.
  - Indice de Calinski-Harabasz est le rapport entre la variance inter-groupes et la variance intra-groupe
  - Le score de distortion ou inertie correspond à la somme des carrés des distances interclusters des observations de leurs centroïdes
- Les scores des métriques suggèrent un k compris entre 2 et 5**

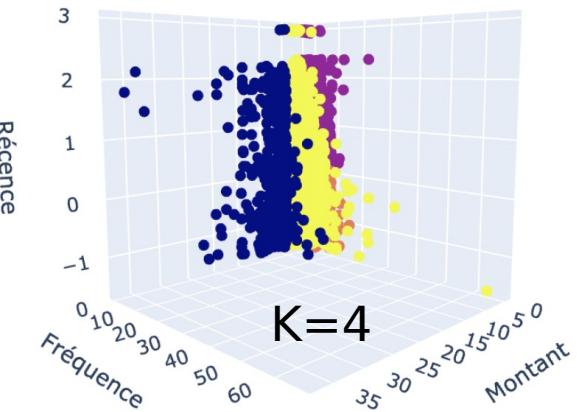
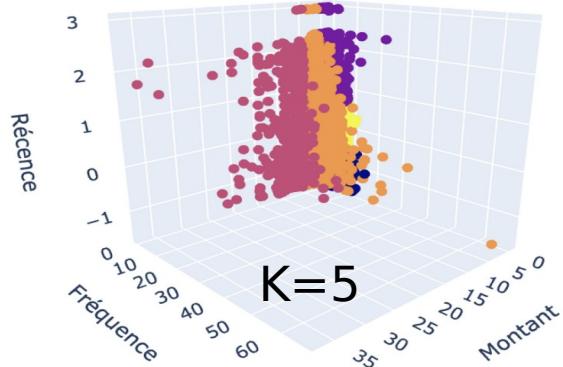
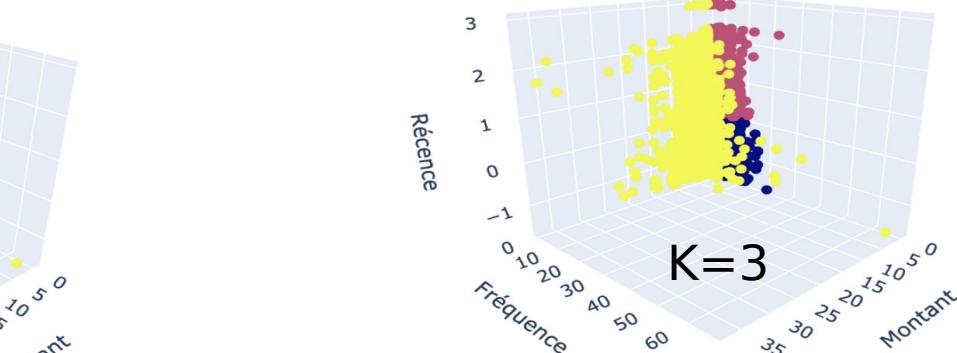
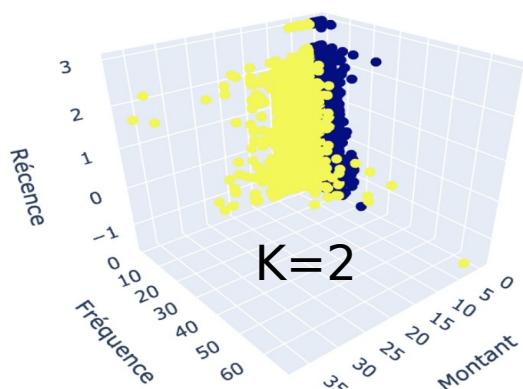
# Partie II : Modèles de clustering

- K-Means : RFM : Analyse multi-variée par boxplot



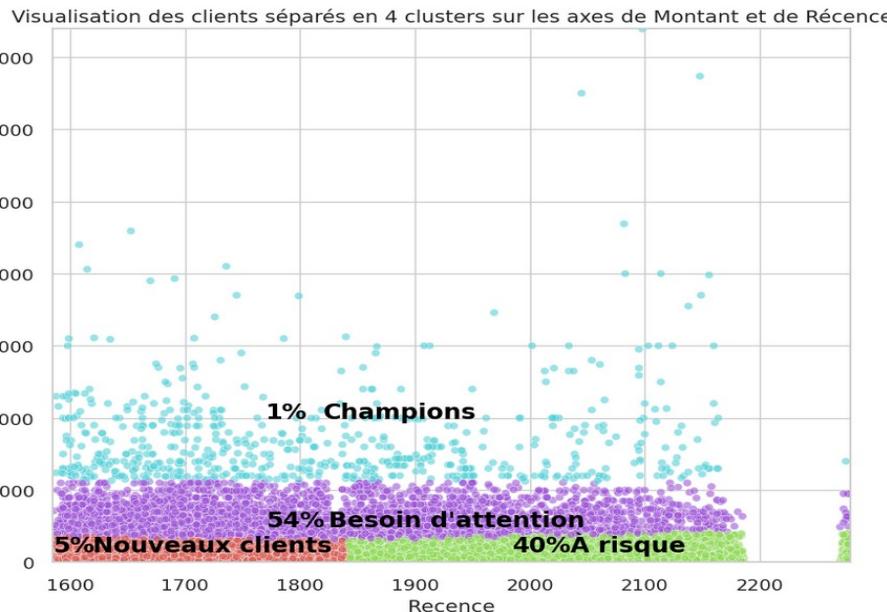
# Partie II : Modèles de clustering

- K-Means : RFM



# Partie II : Modèles de clustering

- K-Means : RFM – Visualisation en 2D des clusters

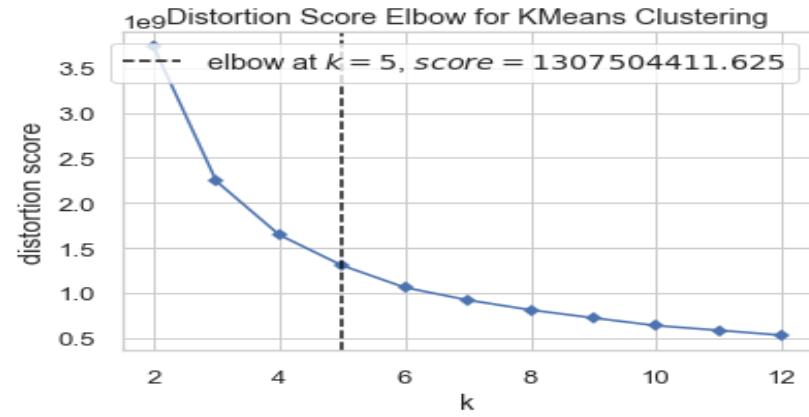
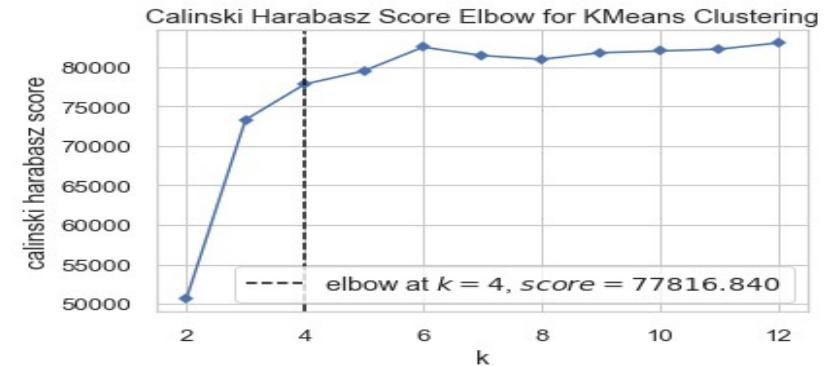
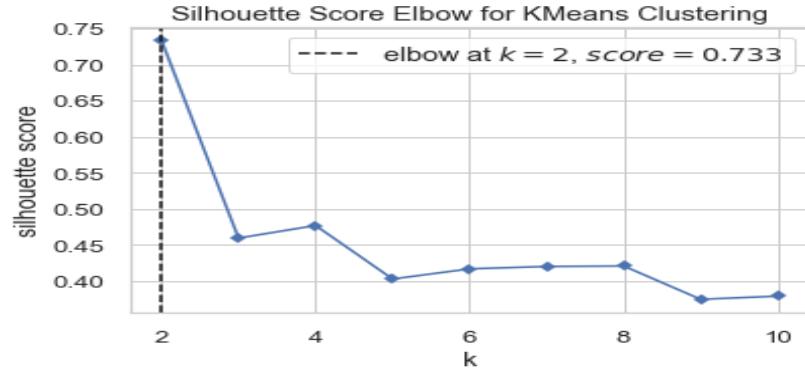


- Nouveaux** : clients qui ont acheté récemment avec un montant faible.
- À risque** : clients qui ont acheté il y-a longtemps avec un montant faible.
- Besoin d'attention** : correspondent aux clients qui ont acheté avec un montant moyen.
- Champions** : clients qui ont acheté avec un montant élevé.

	Montant	Fréquence	Récence
labels_c4			
0	92.936820	1.030525	1347.536801
1	94.500355	1.025947	1608.717647
2	1698.098591	1.070465	1464.500750
3	548.260928	1.111068	1448.631333

# Partie II : Modèles de clustering

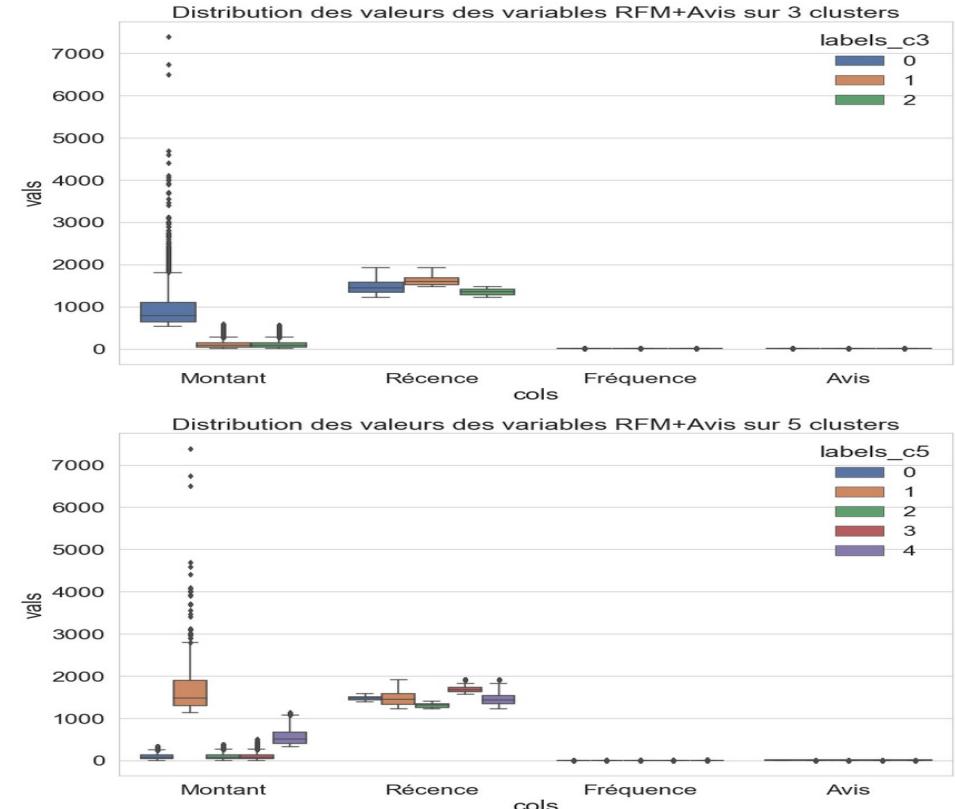
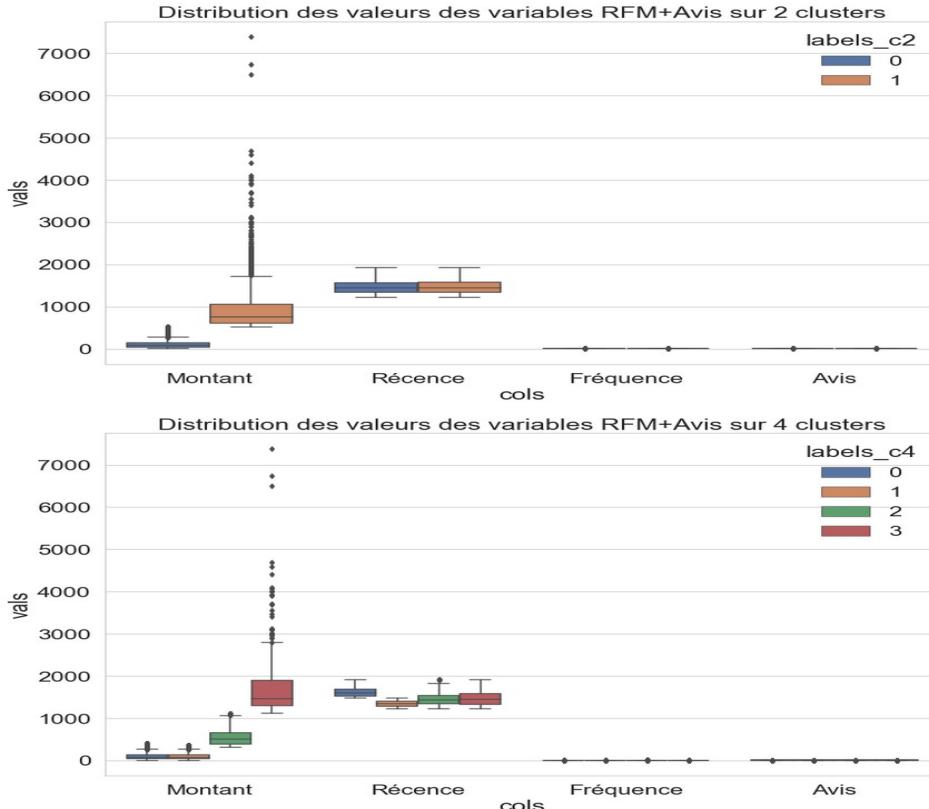
- K-Means : RFM + Avis : On introduit la variable Avis dans le calcul des clusters



# Partie II : Modèles de clustering

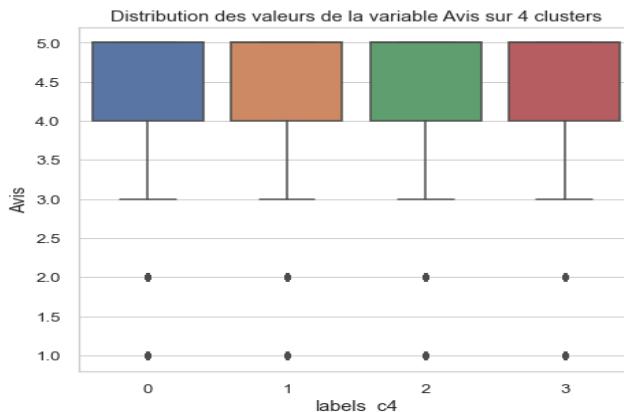
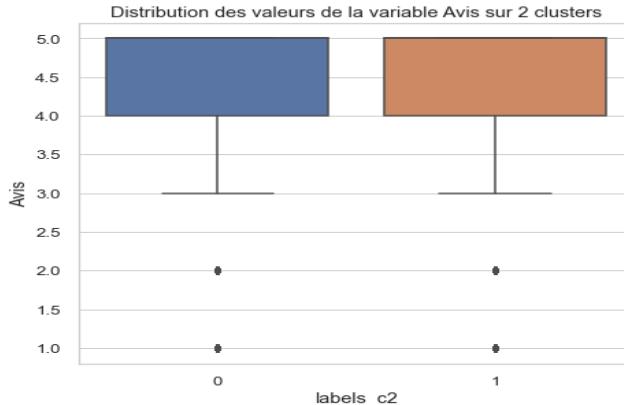
- K-Means : RFM + Avis

## Analyse multivariées par boxplot

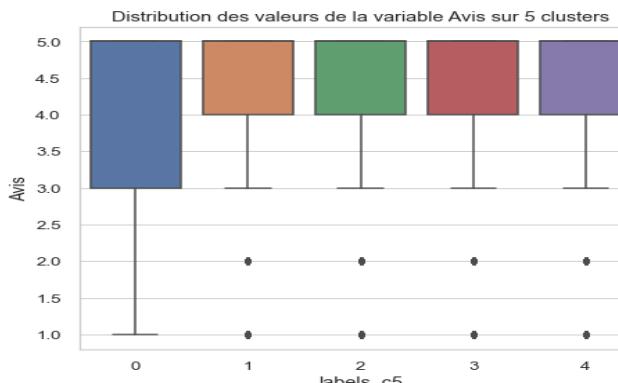
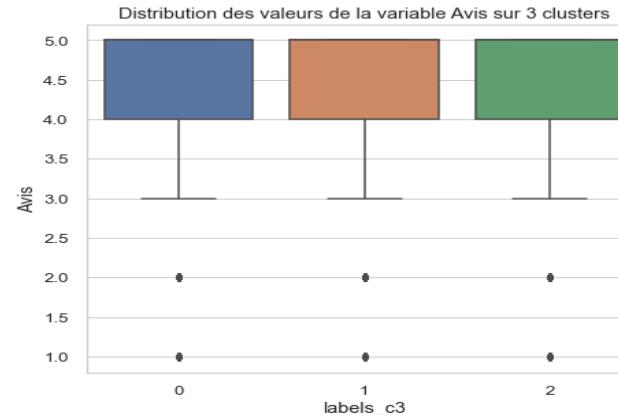


# Partie II : Modèles de clustering

- K-Means : RFM + Avis



## Analyse multivariées par boxplot



- Entre  $k=2$  et  $5$ , seulement le clustering  $k=5$  nous permet de différencier un peu la variabilité des avis des clients.

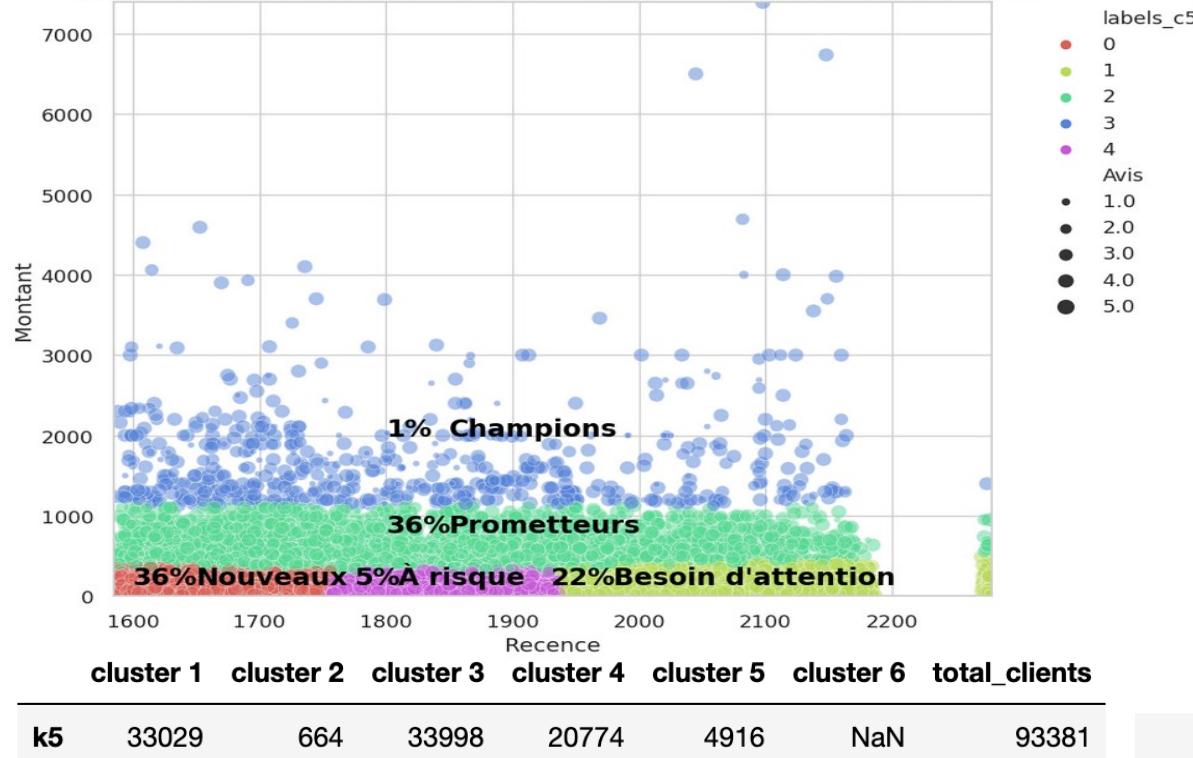
- Le cluster  $k=5$  est retenu pour une segmentation qui tient compte de l'avis des clients.

# Partie II : Modèles de clustering

- K-Means : RFM + Avis

## Visualisation en 2D des clusters

Visualisation des clients séparés en 5 clusters sur les axes de Montant et de Récence



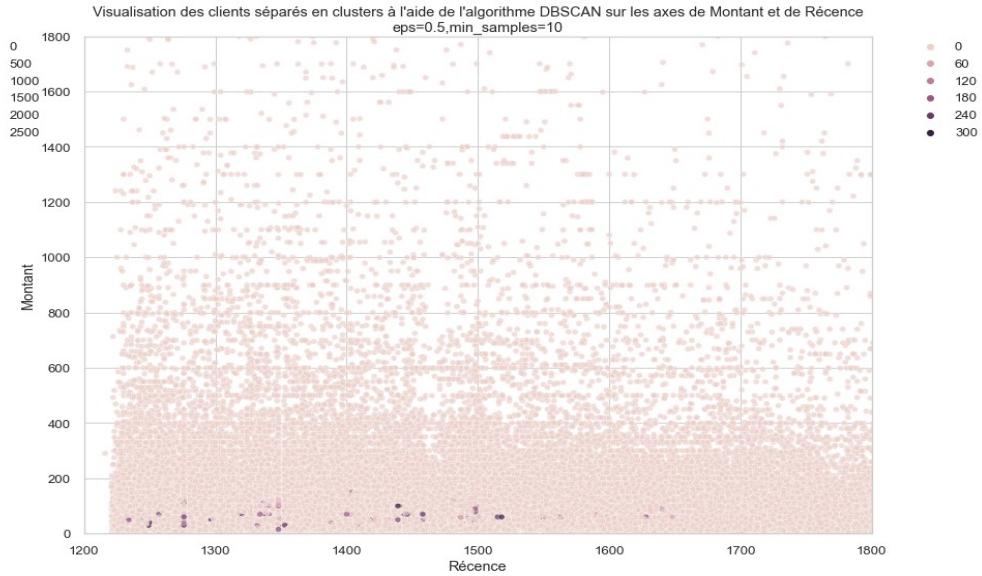
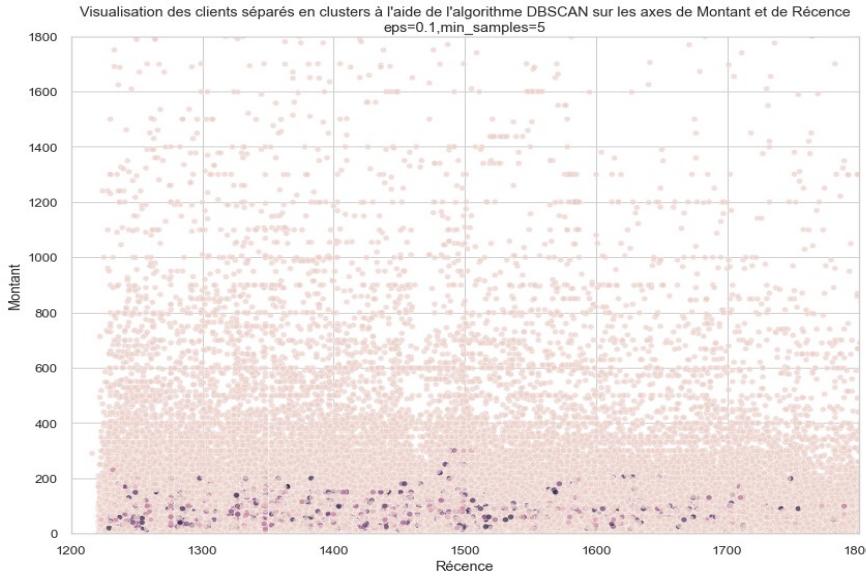
- Prometteurs** : clients qui ont acheté il y-a peu de temps avec un montant faible --> cluster qui contient les clients les moins contents.

	Montant	Fréquence	Récence	Avis
<b>s_c5</b>				
0	90.765951	1.027794	1475.679948	3.990675
1	1700.677364	1.066265	1464.795181	4.000000
2	95.693408	1.031708	1306.962880	4.191864
3	97.361540	1.025561	1679.360017	4.159671
4	557.562720	1.111676	1448.519731	4.033971

# Partie II : Modèles de clustering

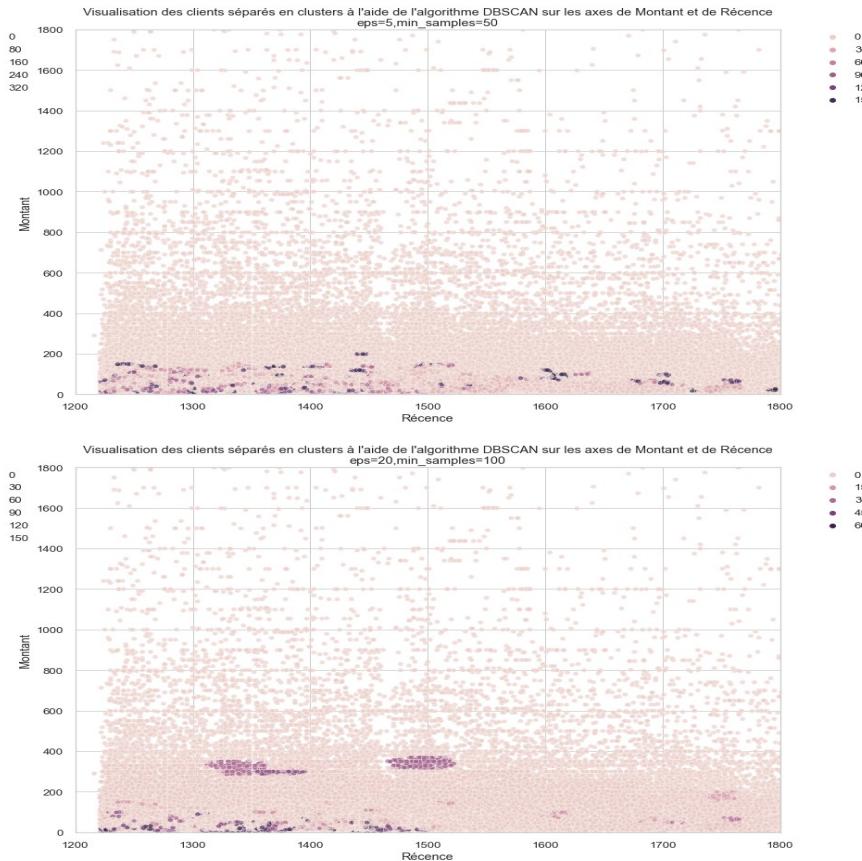
- **DBSCAN**

- l'algorithme DBSCAN considère les clusters comme des zones de haute densité séparées par des zones de faible densité et donc s'appuie sur la densité estimée des clusters pour effectuer le partitionnement.
- Je vais évaluer l'algorithme DBSCAN en modifiant les hyperparamètres **eps** et **min\_samples**



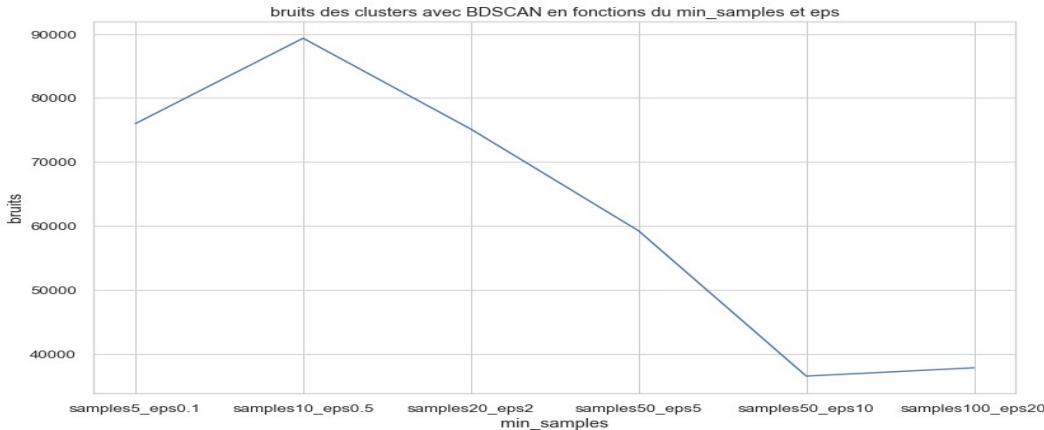
# Partie II : Modèles de clustering

## DBSCAN



# Partie II : Modèles de clustering

- DBSCAN



	DBSCAN1	DBSCAN2	DBSCAN3	DBSCAN4	DBSCAN5	DBSCAN6	Non_clustered
N1	17388.0	NaN	NaN	NaN	NaN	NaN	75993
N2	NaN	4041.0	NaN	NaN	NaN	NaN	89340
N3	NaN	NaN	18239.0	NaN	NaN	NaN	75142
N4	NaN	NaN	NaN	34121.0	NaN	NaN	59260
N5	NaN	NaN	NaN	NaN	56828.0	NaN	36553
N6	NaN	NaN	NaN	NaN	NaN	55533.0	37848

**L'algorithme DBSCAN n'est pas adapté à notre problème de segmentation puisqu'il permet d'exclure beaucoup de clients de la segmentation**

## Partie II : Modèles de clustering

- Clustering hiérarchique agglomératif

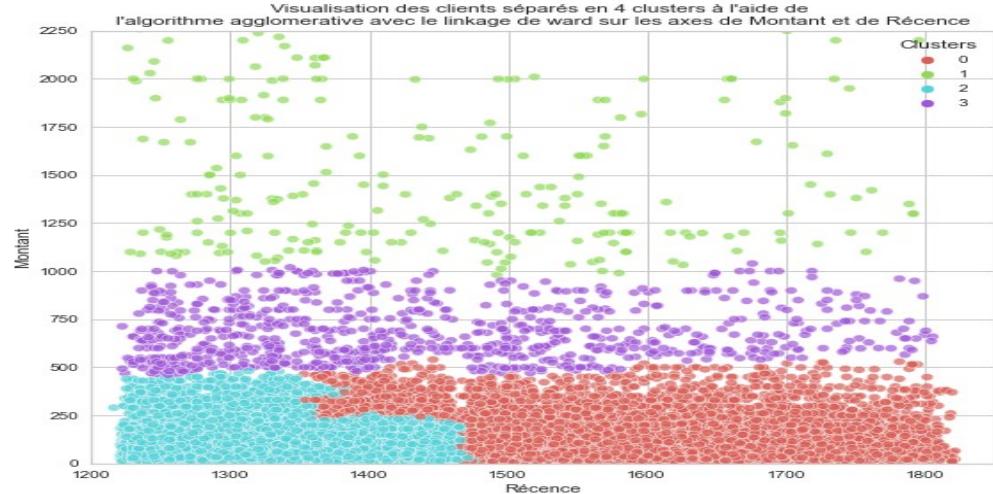
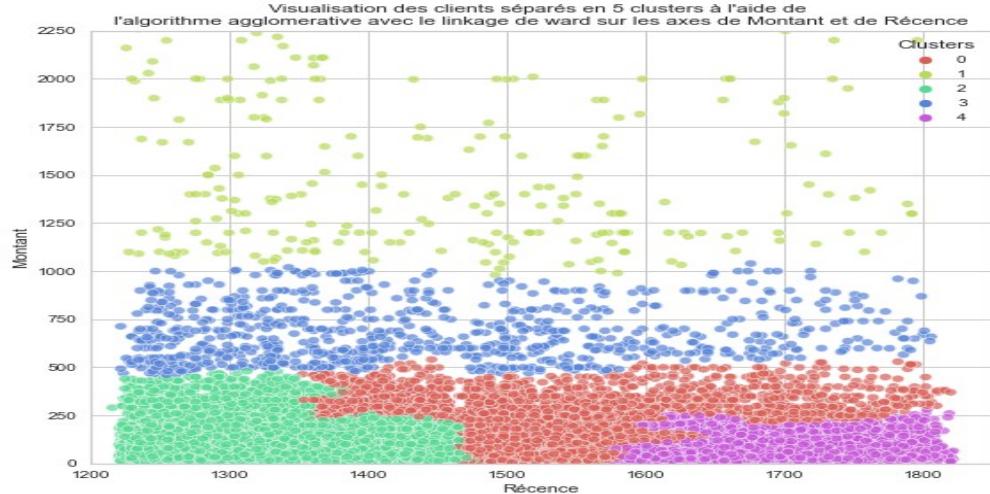
- Dans le cas du clustering agglomératif, d'abord chaque point est un cluster à lui tout seul. Ensuite, on trouve les deux clusters les plus proches, et on les agglomère en un seul cluster. Nous pouvons spécifier le nombre de clusters désiré avec le paramètre **n\_cluster**

Je vais utiliser 2 types de clustering :

- **Ward**: permet l'agrégation des clusters de sorte à minimiser l'augmentation de la variance inter-cluster ou inertie.
- **Complete**: permet l'agrégation des clusters de sorte que la distance entre deux clusters est celle entre les deux points les plus éloignés.

# Partie II : Modèles de clustering

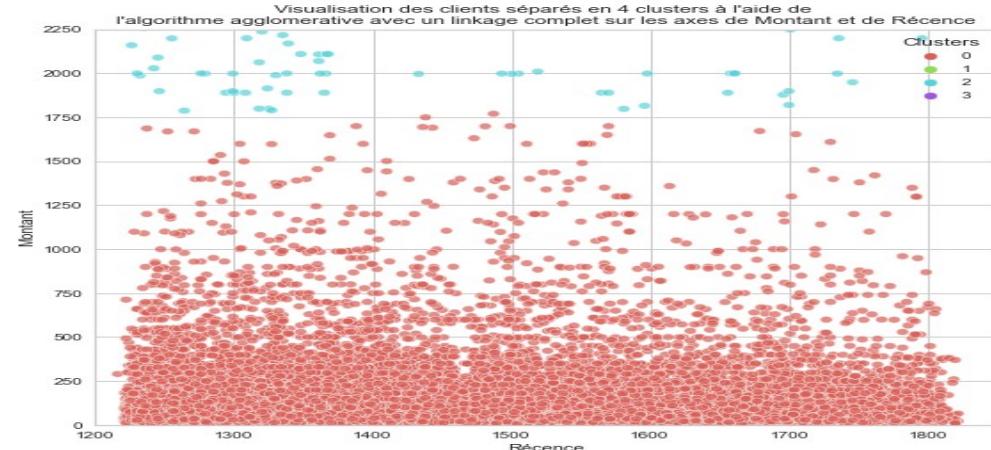
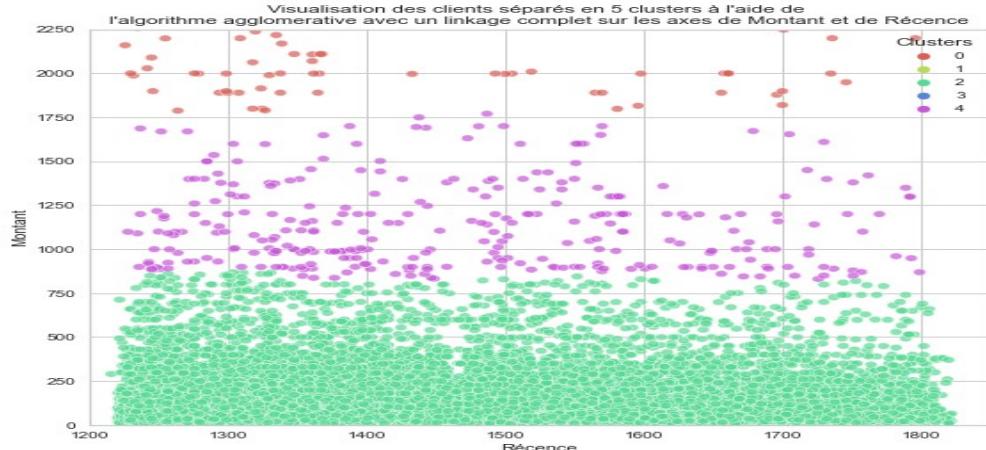
- Clustering hiérarchique agglomératif : Ward linkage



- Les résultats sont proches de K-Means.
- Nombres de clients très déséquilibrés entre les clusters.

# Partie II : Modèles de clustering

- Clustering hiérarchique agglomératif : Complete linkage



	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	total_clients
c4complete	29917	6	76	1	NaN	30000
c5complete	76	6	29605	1	312.0	30000
c4ward	13093	261	15757	889	NaN	30000
c5ward	7635	261	15757	889	5458.0	30000

- Concentration d'environ 99% des clients dans un seul cluster.
- La méthode de ward présente une meilleure dispersion des clients dans les clusters par rapport au complete-linkage clustering.

# Partie III : Maintenance

- Maintenance

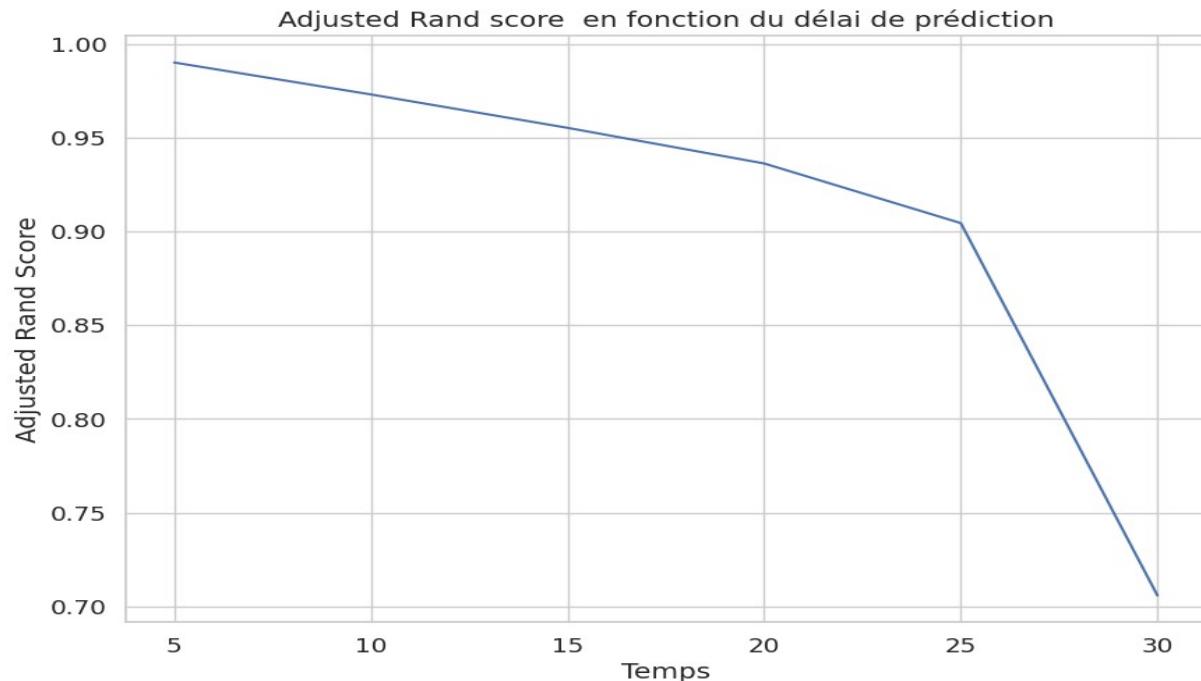
La date d'achat la plus récente 2018-09-03 correspond à un nombre de jours écoulé = 1216 jours et la date d'achat la plus ancienne 2016-10-04 correspond à un nombre de jours écoulé = 1915 jours.



- La base de données est très variable en fonction du temps.
- Créer un premier fichier initial de clients sans les derniers 45 jours.
- Faire une simulation pour prédire les clusters au fur et à mesure du temps avec un délai de 5 jours.

# Partie III : Maintenance

- Maintenance



- Le Rand score ajusté est un calcul qui permet d'évaluer la similarité entre les clusters prédits et vrais
- Un score proche de 0 est donné pour un étiquetage aléatoire
- un score = 1 clusterings identiques

	Temps	Adjusted Rand Score
0	5	0.990351
1	10	0.973301
2	15	0.955471
3	20	0.936444
4	25	0.904599
5	30	0.705727

D'après les résultats du Rand score il est conseillé de faire une maintenance du clustering tout les 25 jours environ.

# Conclusion

- ❖ 3 méthodes de clustering testés.
- ❖ Choix de nombre de clusters en se basant sur les tests statistiques et des règles métier (méthode RFM).
- ❖ Les résultats de clustering obtenus avec le K-Means sont assez similaires avec les résultats de l'algorithme hiérarchiques de Ward.
- ❖ Le DBSCAN n'est pas adapté à notre jeu de données (fonction par densité).
- ❖ Un contrat peut être proposé afin de réaliser une maintenance de la base de données tout les 20 jours et ce qui permet d'assurer la stabilité des segments avec le temps.