PROJET 5
Olist Segmentation Client
modélisation ML non supervisée

Catherine BRICE
Openclassrooms formation Data Scientist
Le 03/02/2022

Plan du document

- Introduction
- Cleaning et analyse exploratoire des données
- Pistes de modélisation et modèle final
- Contrat de maintenance

Introduction

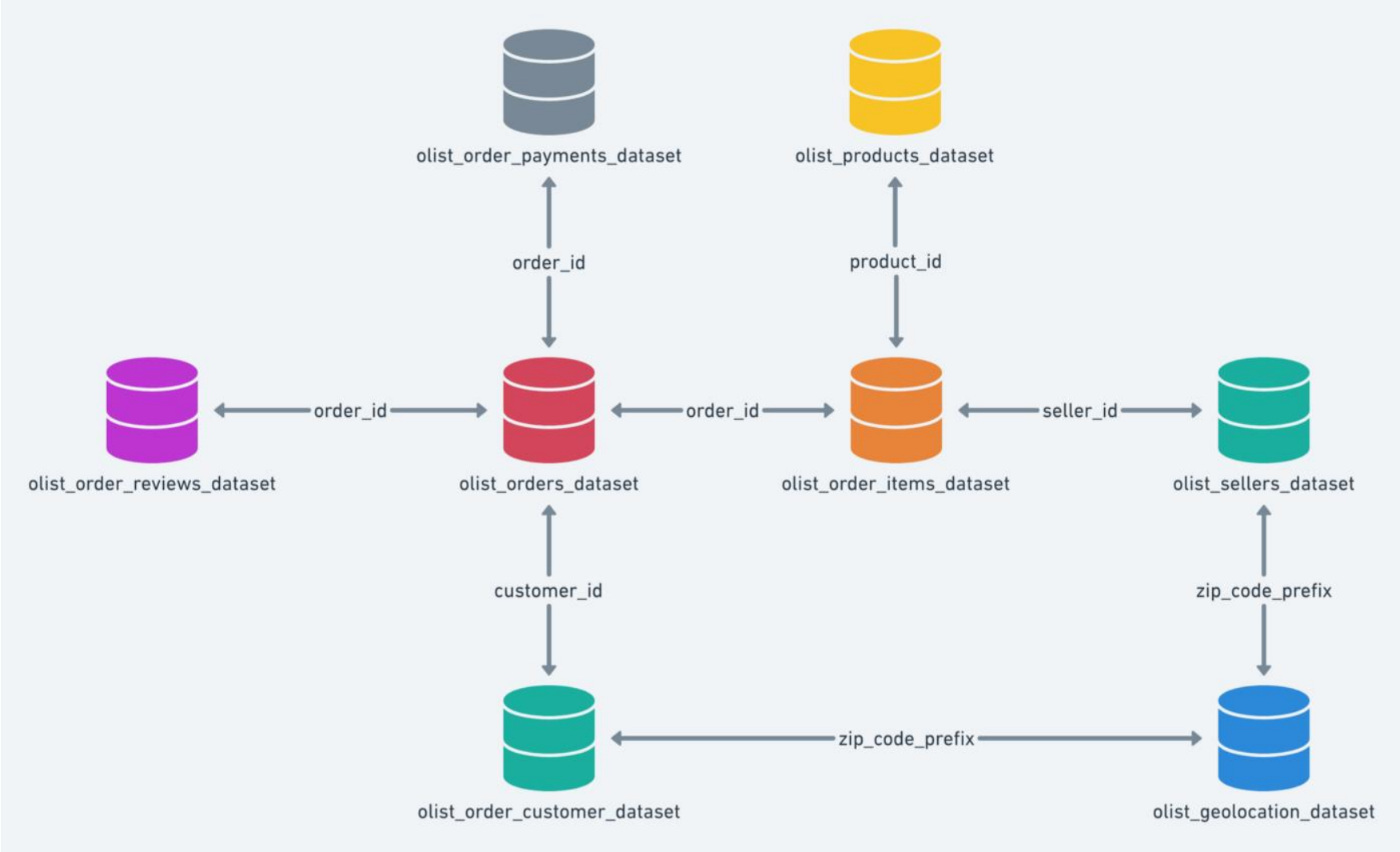
Olist : plateforme brésilienne de e-commerce

Buts du projet :

- Comprendre ses utilisateurs d'un point de vue marketing
- Segmenter les clients par modélisation non supervisée
- Proposer un contrat de maintenance afin de mieux orienter la stratégie marketing

Les datas issues de Olist

A partir d'une base de données anonymisée de Olist, historique de commandes, produits achetés, commentaires de satisfaction, localisation des clients de 2016 à 2018



Les dataframes : shapes

Dataframes	Shapes
customers	(99441, 5)
geolocation	(1000163, 5)
items	(112650, 7)
payments	(103886, 5)
reviews	(99224, 7)
orders	(99441, 8)
products prod_cat_name_tr	(32951, 9)
sellers	(3095, 4)
products_trans	(71, 2)

Nettoyage des données	Actions
codes postaux clients	supprimer les doublons zipcode, 1 Zip_code = plusieurs latitudes et longitudes, 19608 Zip_code après fusion
supprimer les variables inutiles ou redondantes	'review_comment_title', 'review_comment_message' 'product_photos_qty','product_weight_g','product_length_cm','product_height_cm','product_width_cm' 'payment_sequential' 'product_category_name' 'product_name_lenght' 'product_description_lenght' 'shipping_limit_date', 'geolocation_zip_code_prefix', 'geolocation_city', 'geolocation_state'
sélection des commandes livrées	merged = merged [merged['order_status'] == 'delivered']
valeurs manquantes supprimées	order_approved_at 160 order_delivered_carrier_date 1783 order_delivered_customer_date 2965
accents sur sao paulo à supprimer	
Variables dates à retravailler pd.to_datetime Variables temps (livraison et	merged['order_purchase_timestamp'] merged['order_approved_at'] merged['order_delivered_carrier_date'] merged['order_delivered_customer_date'] = merged['order_estimated_delivery_date'] = merged['shipping_limit_date'] = merged['review_creation_date'] = merged['review_answer_timestamp'] = merged["order_months"] - merged["order_months2"] - merged["order_year"] - merged["order_date"] merged['delivered_time'] = (merged['order_delivered_customer_date'] - merged['order_purchase_timestamp']).dt.days

Fusion des dataframes

```
0]: merged.head()
```

	order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_c
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10-
1	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10-
2	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10-
3	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10-
4	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10-

5 rows × 44 columns

Suppression des NaN : **There is 5.46% nan values**

column	nan %
order_id	0.000000
customer_id	0.000000
order_status	0.000000
order_purchase_timestamp	0.000000
order_estimated_delivery_date	0.000000
payment_type	0.003359
payment_sequential	0.003359
payment_installments	0.003359
payment_value	0.003359
order_approved_at	0.148929
customer_state	0.180282
customer_city	0.180282
customer_zip_code_prefix	0.180282
customer_unique_id	0.180282
geolocation_lat	0.180282
geolocation_state	0.180282
geolocation_city	0.180282
geolocation_zip_code_prefix	0.180282
geolocation_lng	0.180282
order_item_id	0.709371
freight_value	0.709371
product_id	0.709371
seller_id	0.709371
shipping_limit_date	0.709371
price	0.709371
seller_state	0.709371
seller_city	0.709371
seller_zip_code_prefix	0.709371
product_weight_g	0.723928
product_length_cm	0.723928
product_height_cm	0.723928
product_width_cm	0.723928
review_score	0.837584
review_comment_message	0.837584
review_answer_timestamp	0.837584

column	nan %
order_id	0.0
product_name_lenght	0.0
product_description_lenght	0.0
product_photos_qty	0.0
product_weight_g	0.0
product_length_cm	0.0
product_height_cm	0.0
product_width_cm	0.0
product_category_name	0.0
review_answer_timestamp	0.0
seller_zip_code_prefix	0.0
seller_state	0.0
customer_unique_id	0.0
customer_zip_code_prefix	0.0
customer_city	0.0
customer_state	0.0
geolocation_zip_code_prefix	0.0
geolocation_city	0.0
geolocation_state	0.0
seller_city	0.0
review_creation_date	0.0
review_comment_message	0.0
review_comment_title	0.0
customer_id	0.0
order_status	0.0
order_purchase_timestamp	0.0
order_approved_at	0.0
order_delivered_carrier_date	0.0
order_delivered_customer_date	0.0
order_estimated_delivery_date	0.0
order_item_id	0.0
product_id	0.0
seller_id	0.0

Dataframe finale ‘merged’

```
]: merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 169358 entries, 0 to 178608
Data columns (total 36 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   order_id                      169358 non-null  object
 1   customer_id                   169358 non-null  object
 2   order_status                  169358 non-null  object
 3   order_purchase_timestamp      169358 non-null  datetime64[ns]
 4   order_item_id                 169358 non-null  object
 5   product_id                    169358 non-null  object
 6   seller_id                     169358 non-null  object
 7   price                         169358 non-null  float64
 8   freight_value                 169358 non-null  float64
 9   payment_sequential            169358 non-null  float64
 10  payment_type                  169358 non-null  object
 11  payment_installments          169358 non-null  float64
 12  payment_value                 169358 non-null  float64
 13  review_id                     169358 non-null  object
 14  review_score                  169358 non-null  float64
 15  review_comment_title          169358 non-null  object
 16  review_comment_message        169358 non-null  object
 17  product_weight_g              169358 non-null  float64
 18  product_category_name         169358 non-null  object
 19  seller_zip_code_prefix        169358 non-null  object
 20  seller_city                   169358 non-null  object
 21  seller_state                  169358 non-null  object
 22  customer_unique_id            169358 non-null  object
 23  customer_zip_code_prefix      169358 non-null  object
 24  customer_city                 169358 non-null  object
 25  customer_state                169358 non-null  object
 26  geolocation_lat               169358 non-null  float64
```

Analyse des données

91211

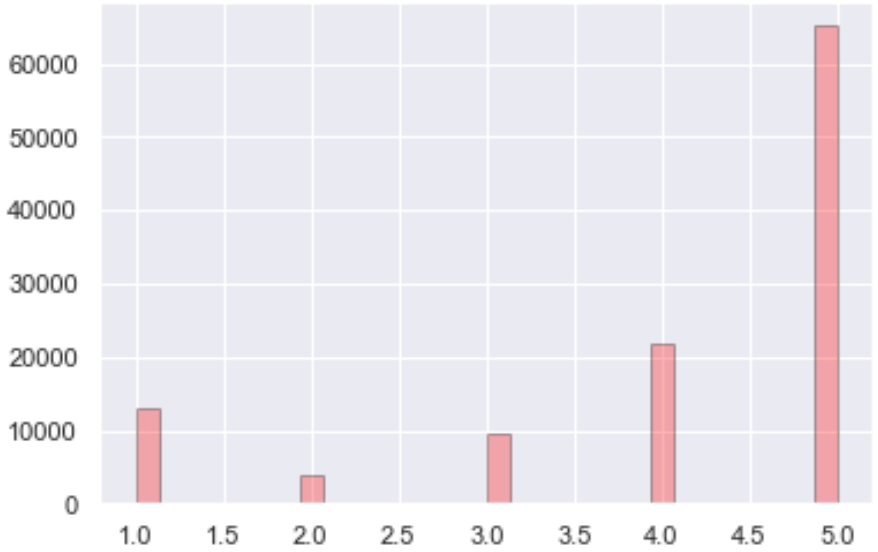
94479

27

71

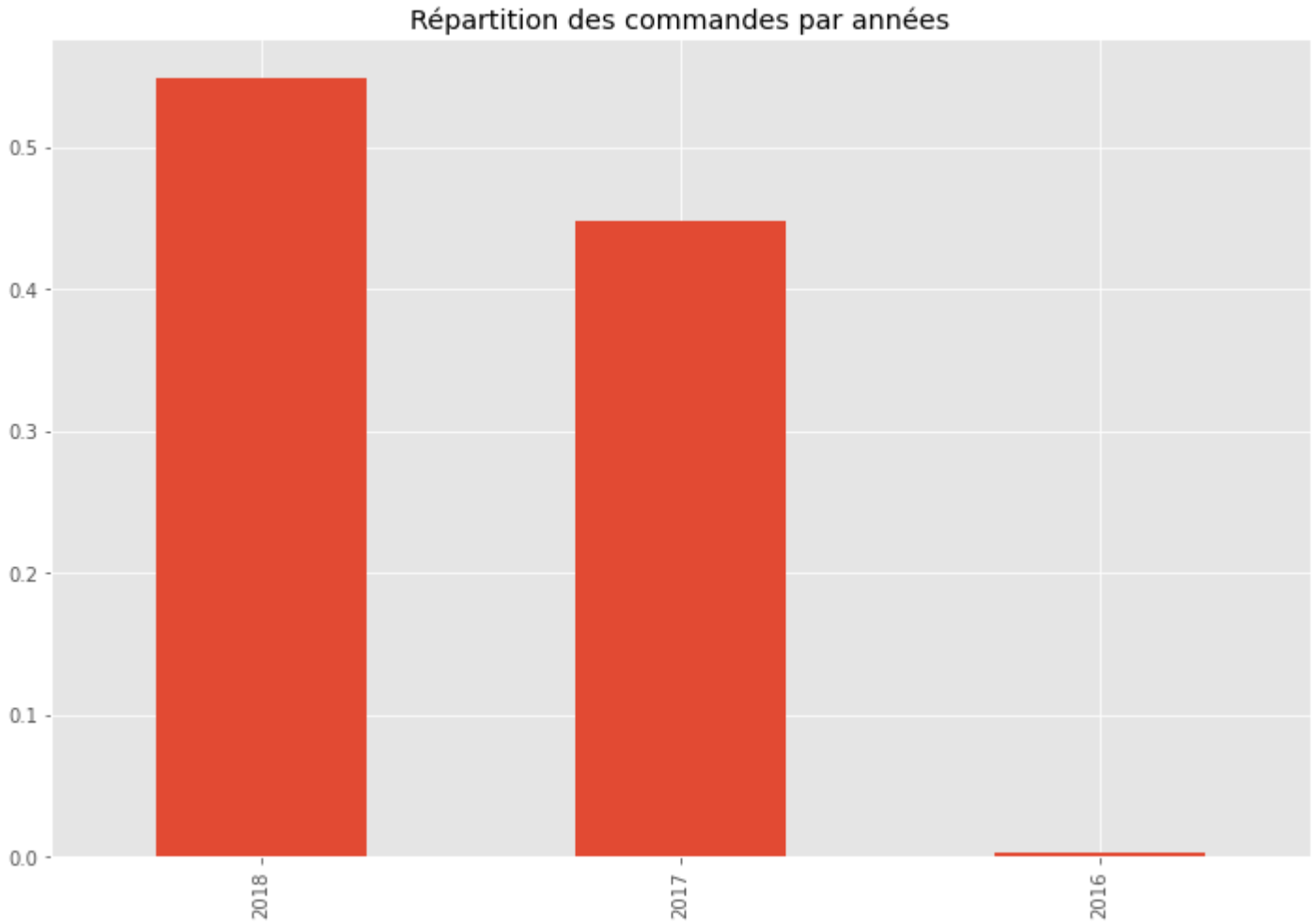
14678

2907

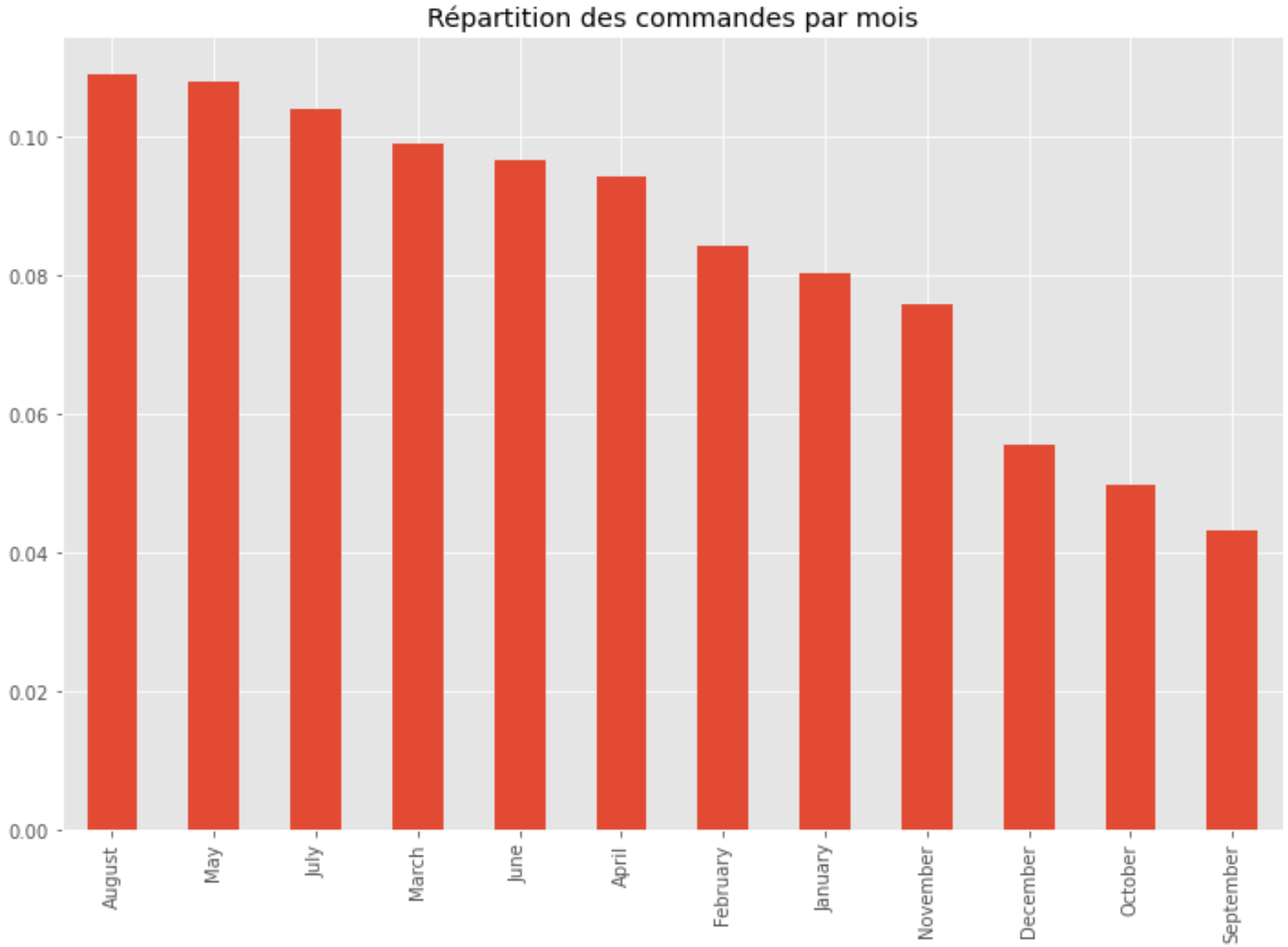


Satisfaction client de 1 à 5

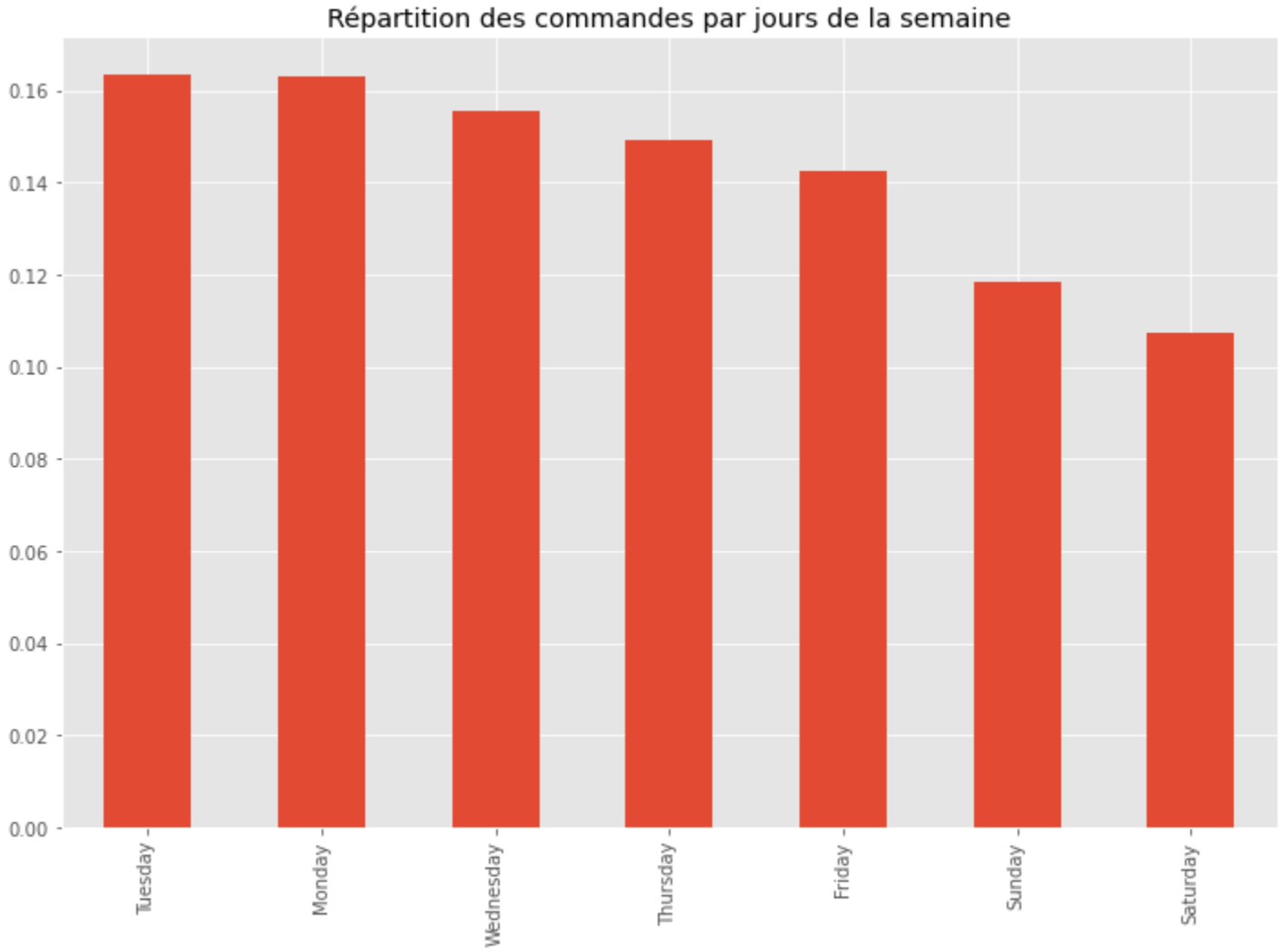
Les années de vente



Les mois d’achats clients

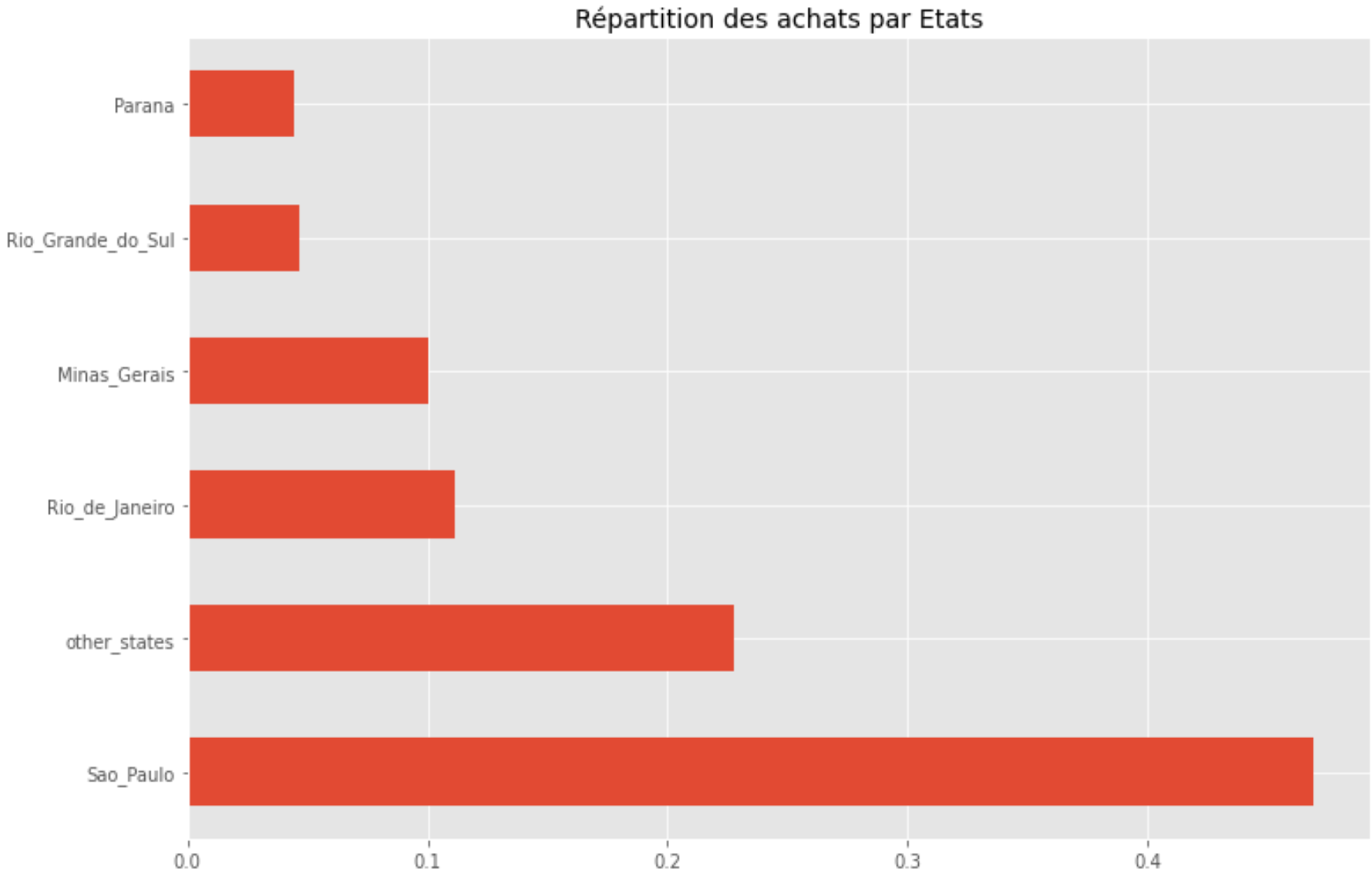


Les jours d’achats clients



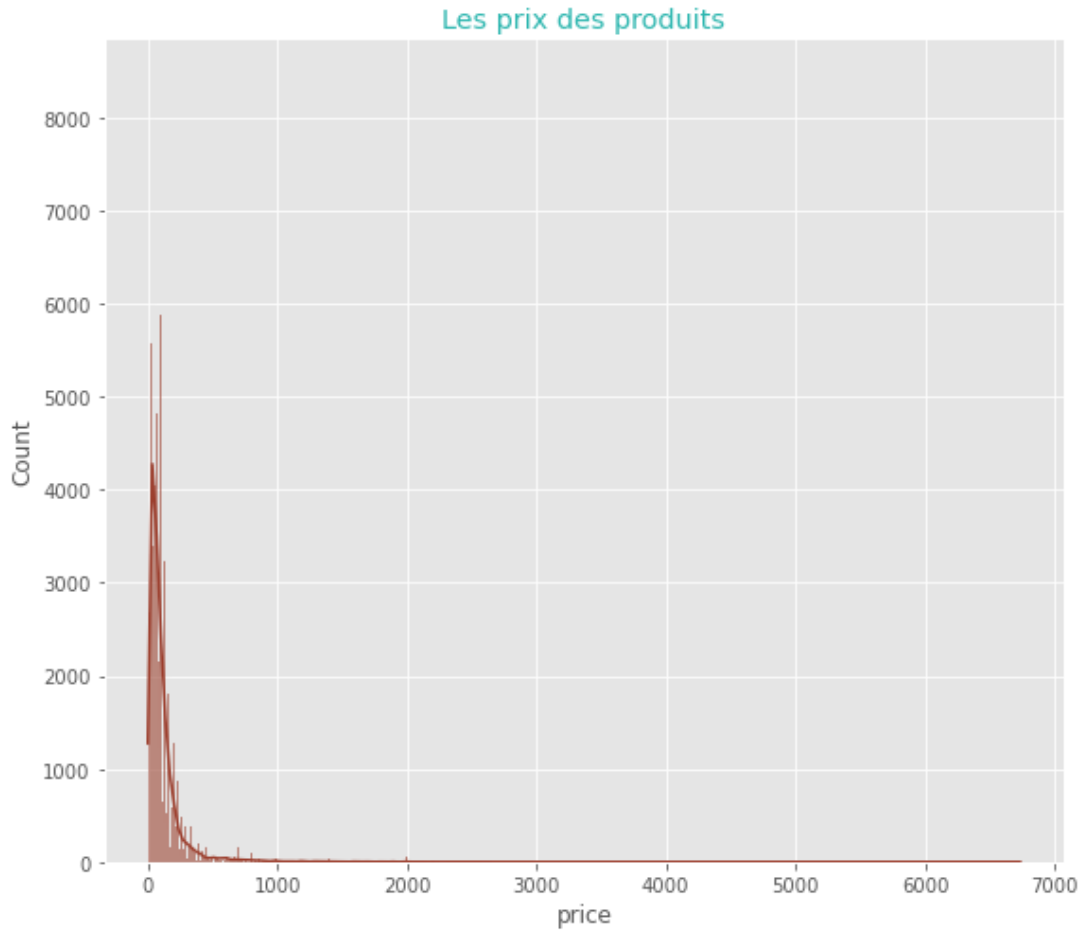
Concentration des clients

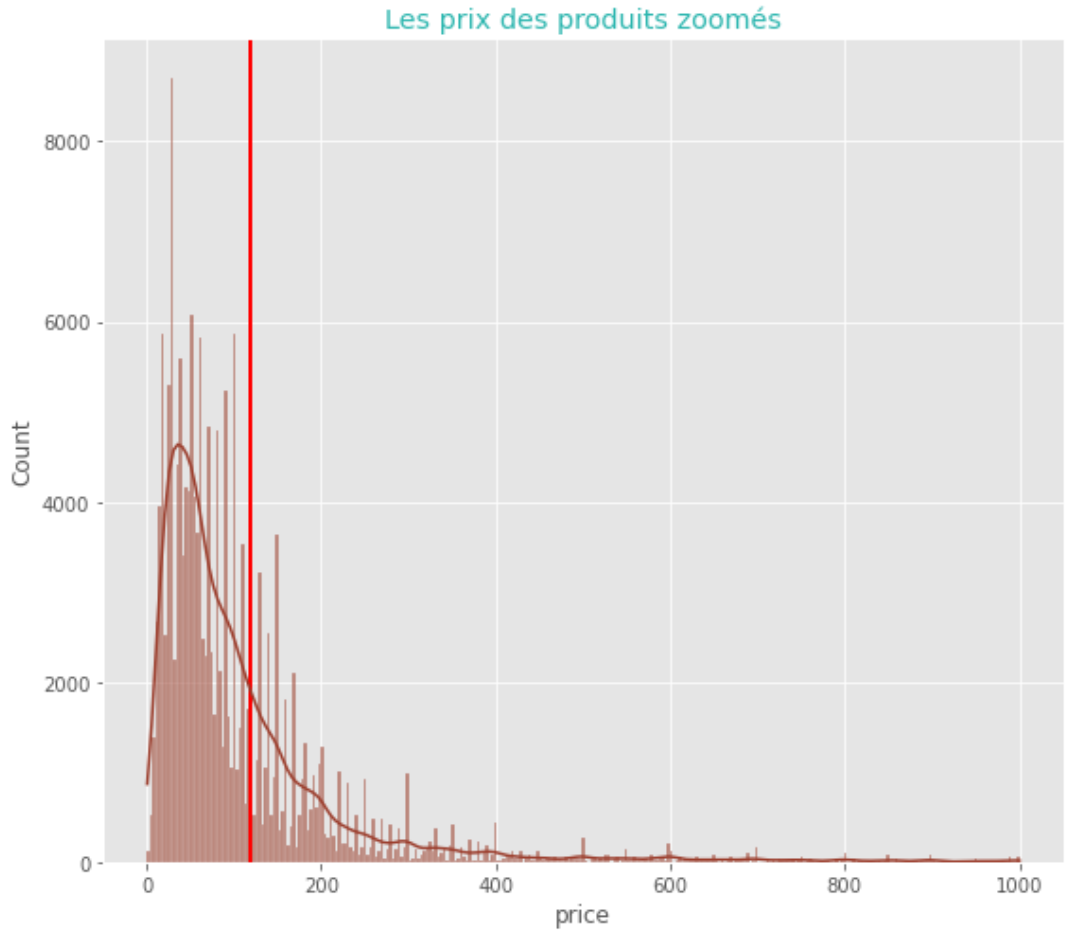
6 grands groupes d’états acheteurs



Les prix des produits

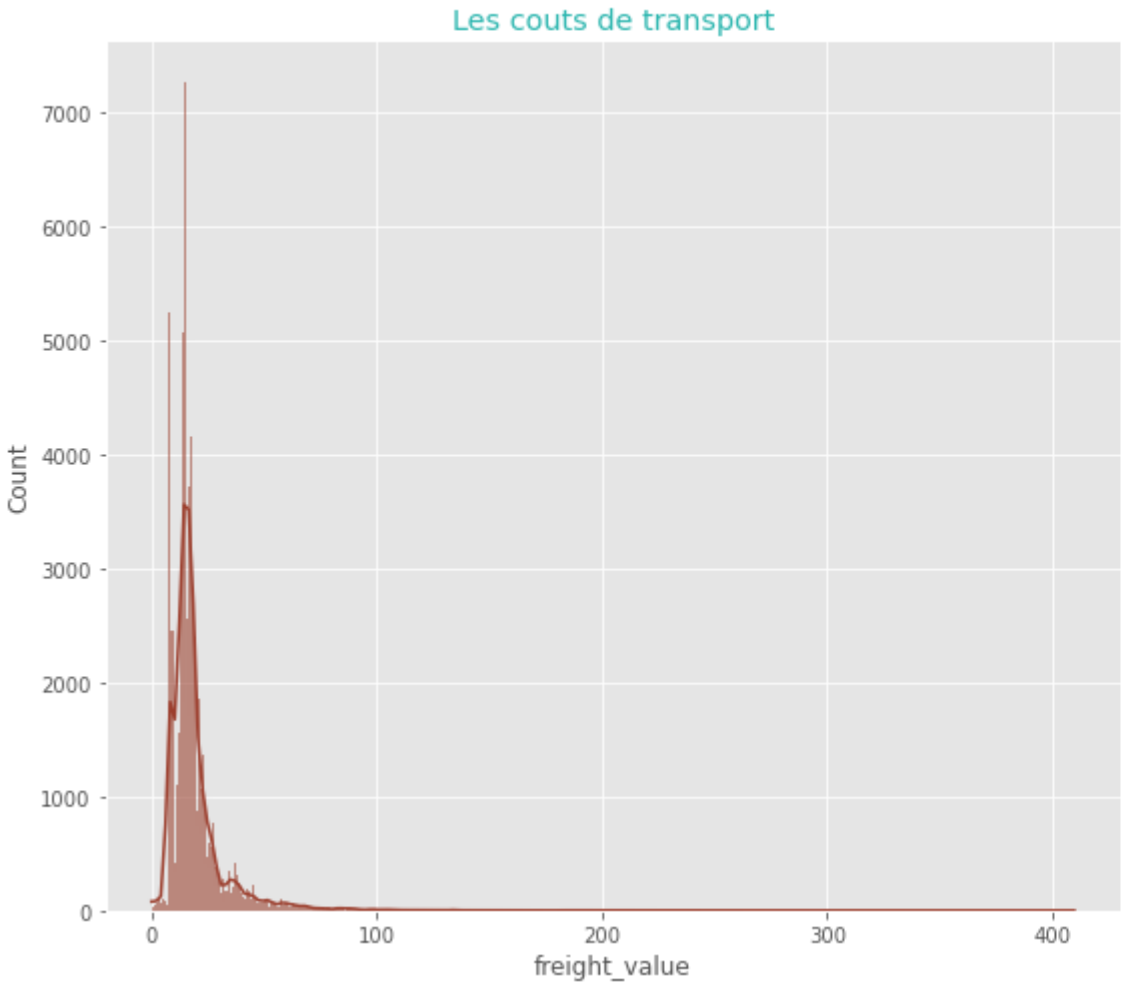
Distribution des prix des produits

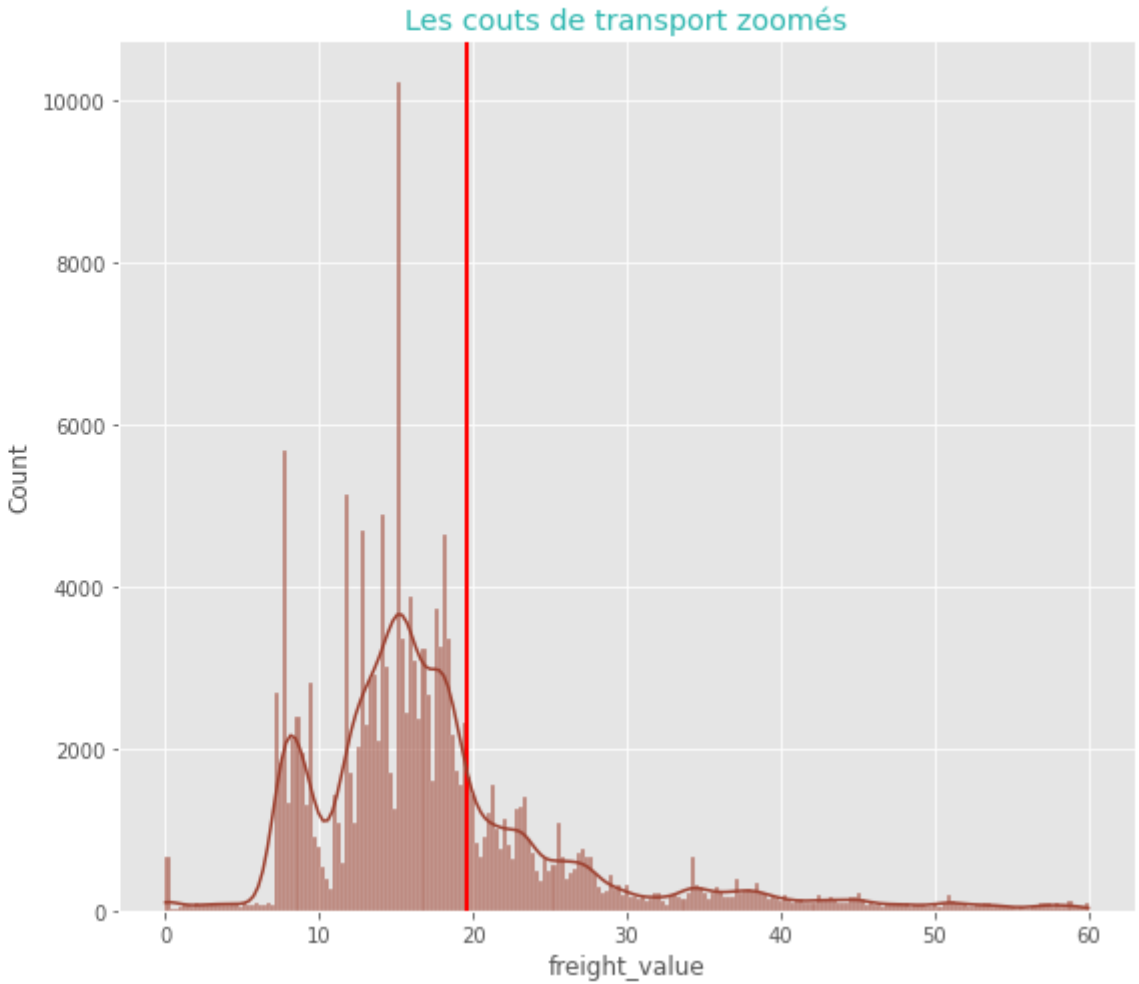




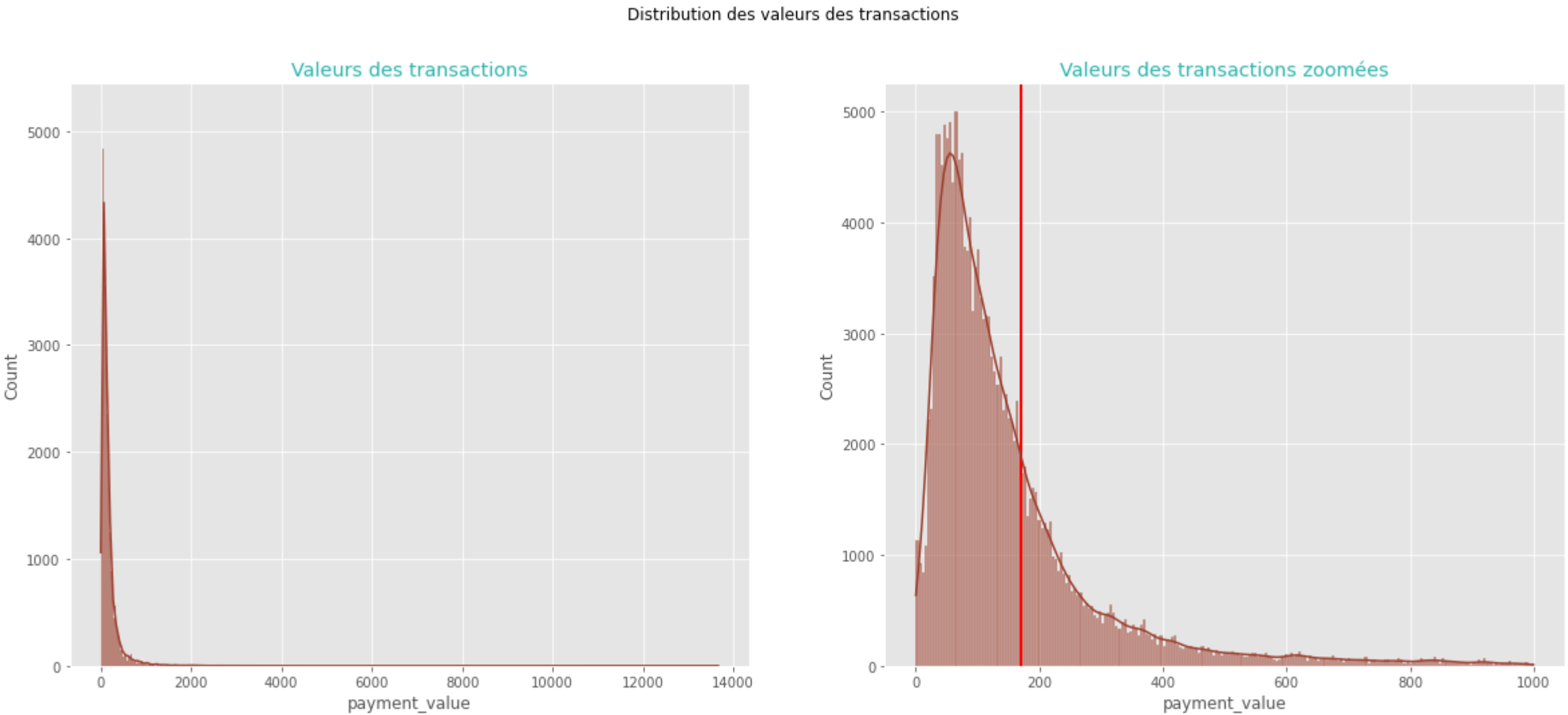
Le cout du freight

Distribution des couts de transport

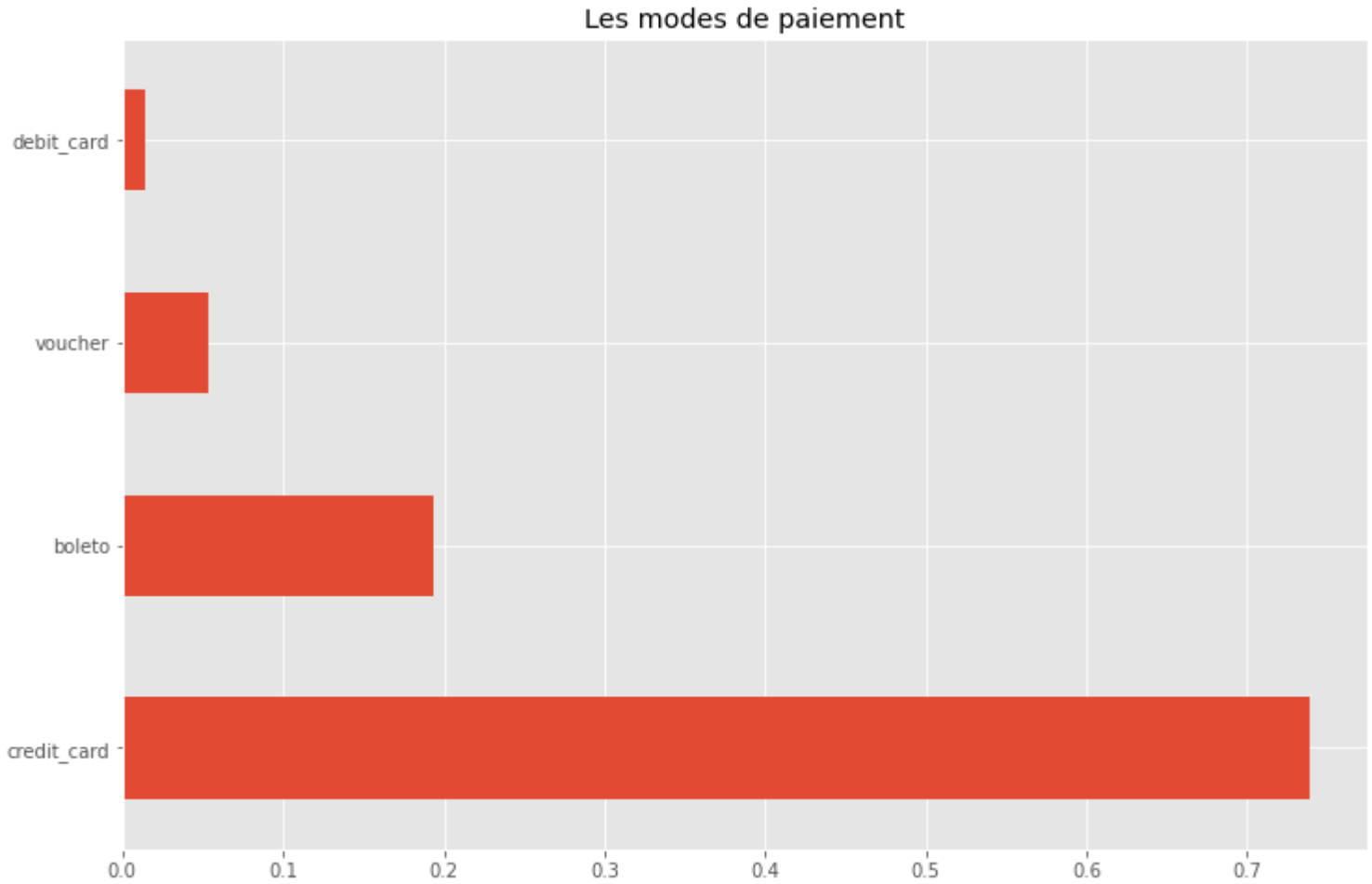




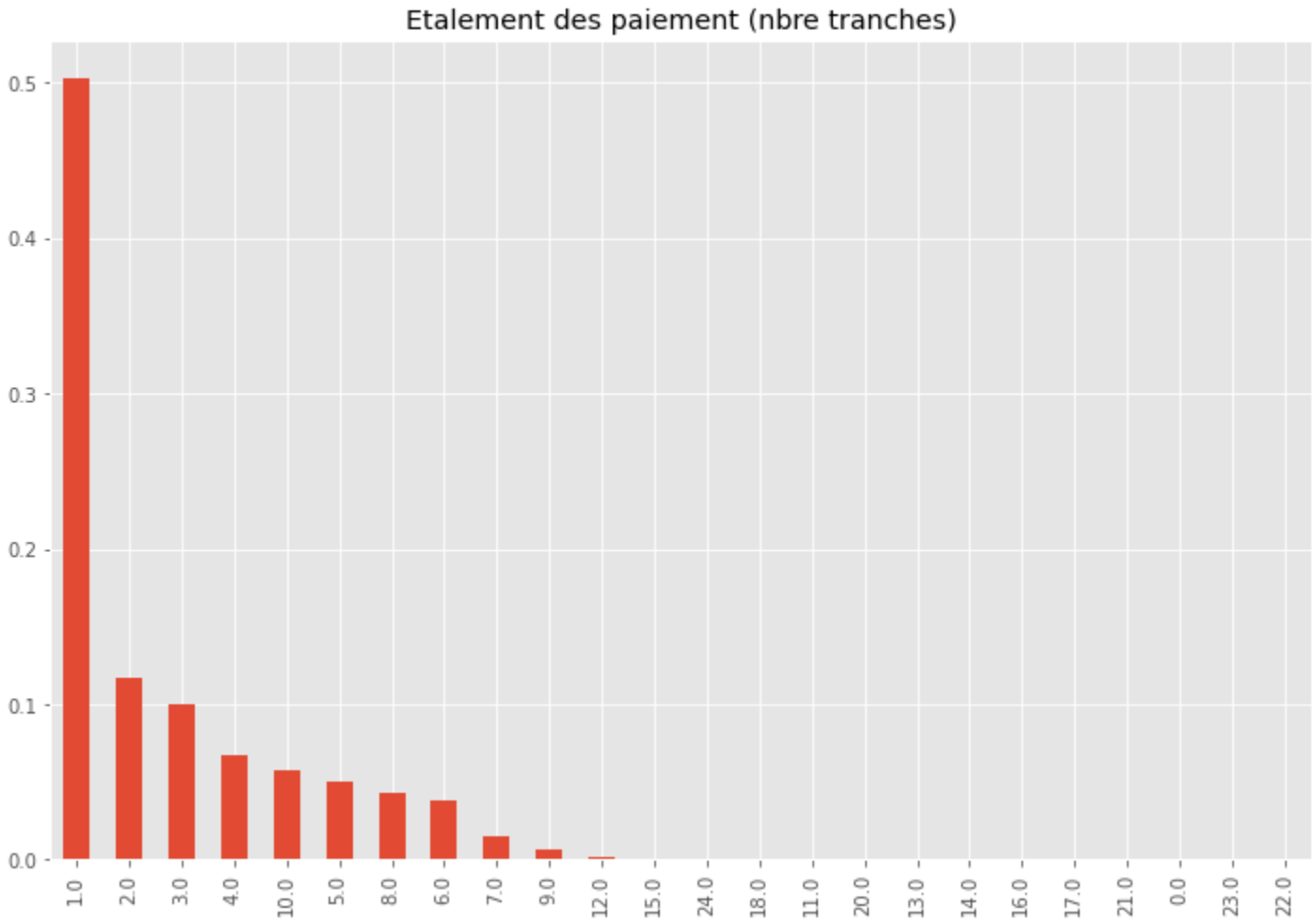
Les valeurs de transactions



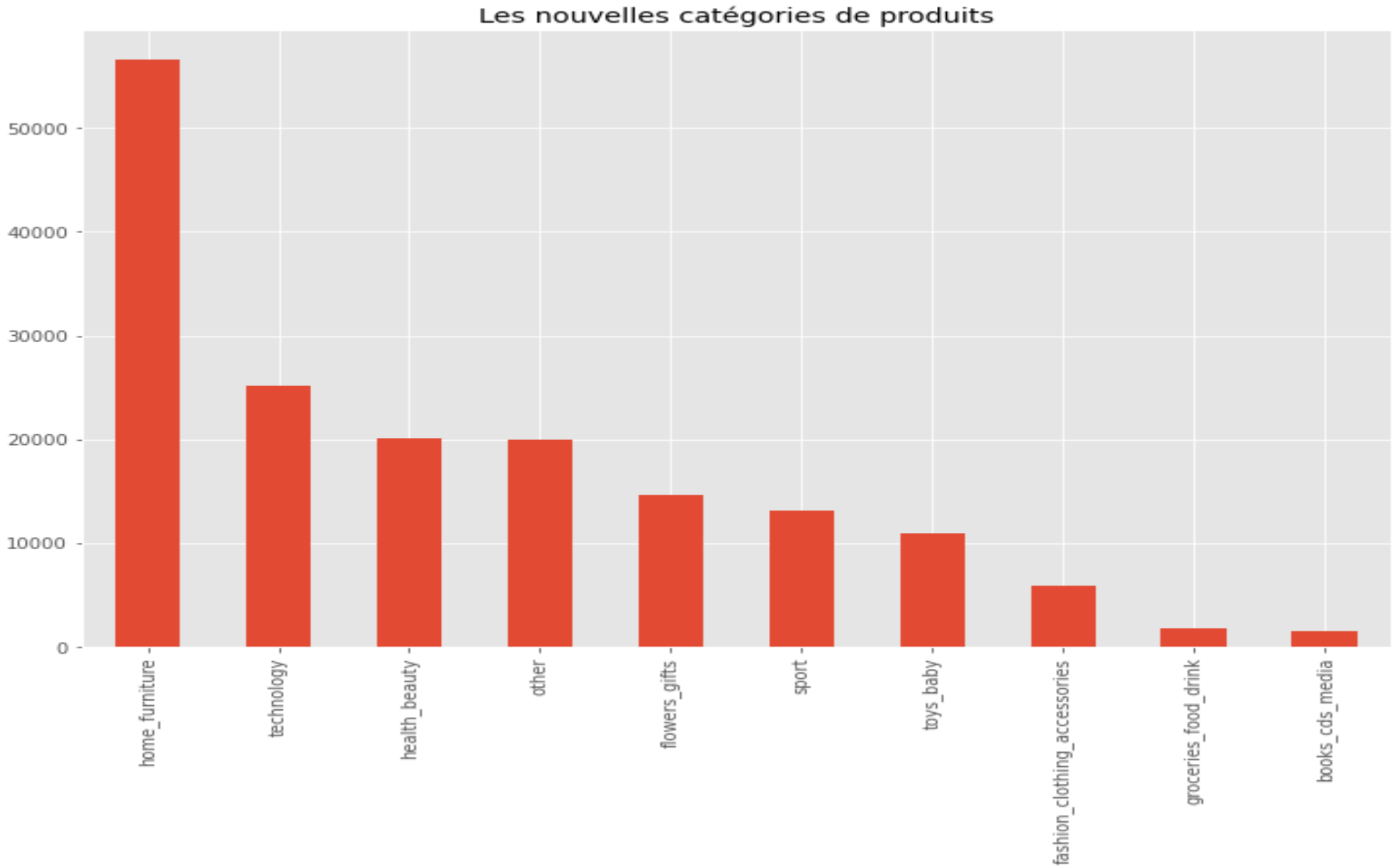
Les moyens de paiement



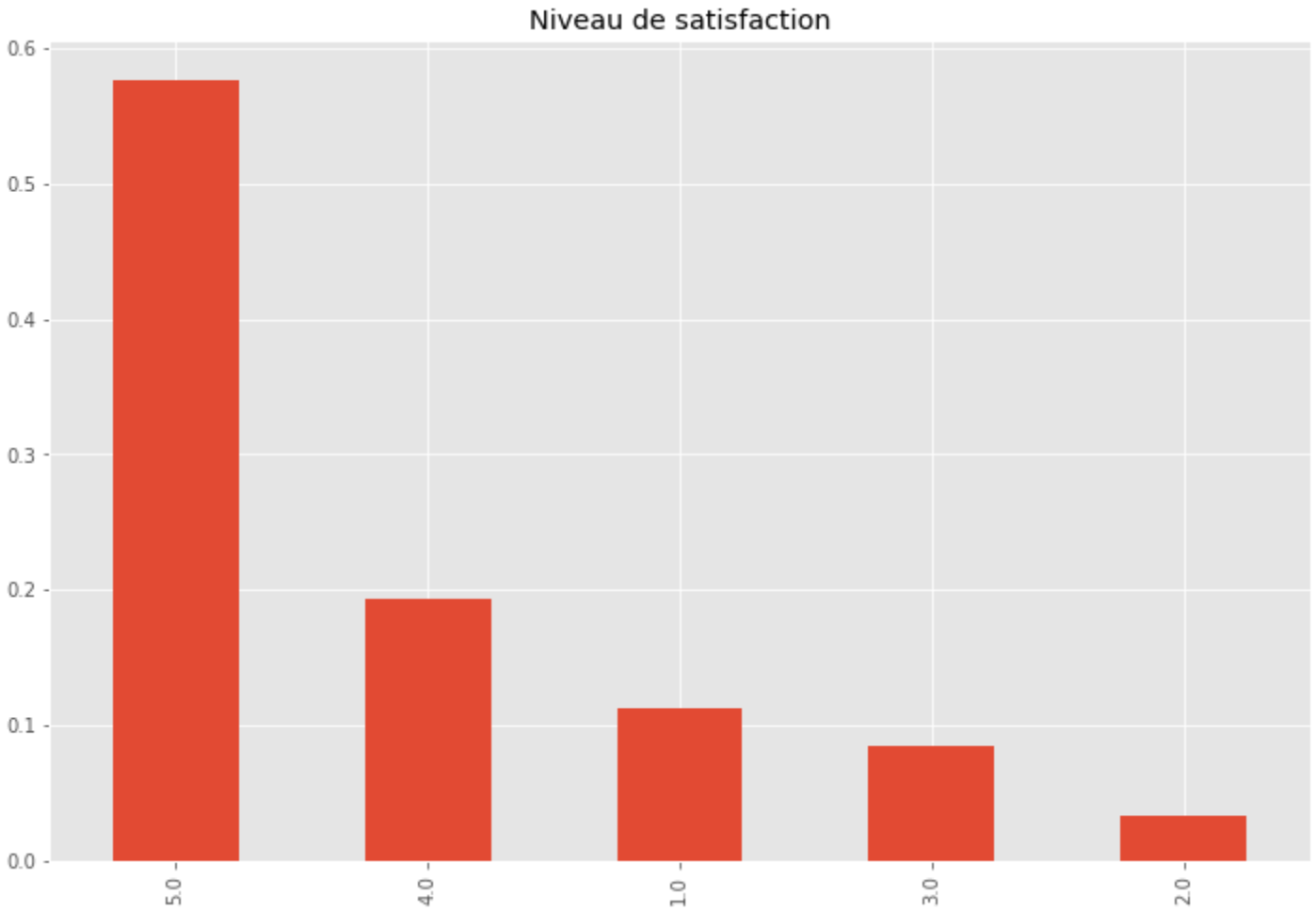
Etalement des paiements



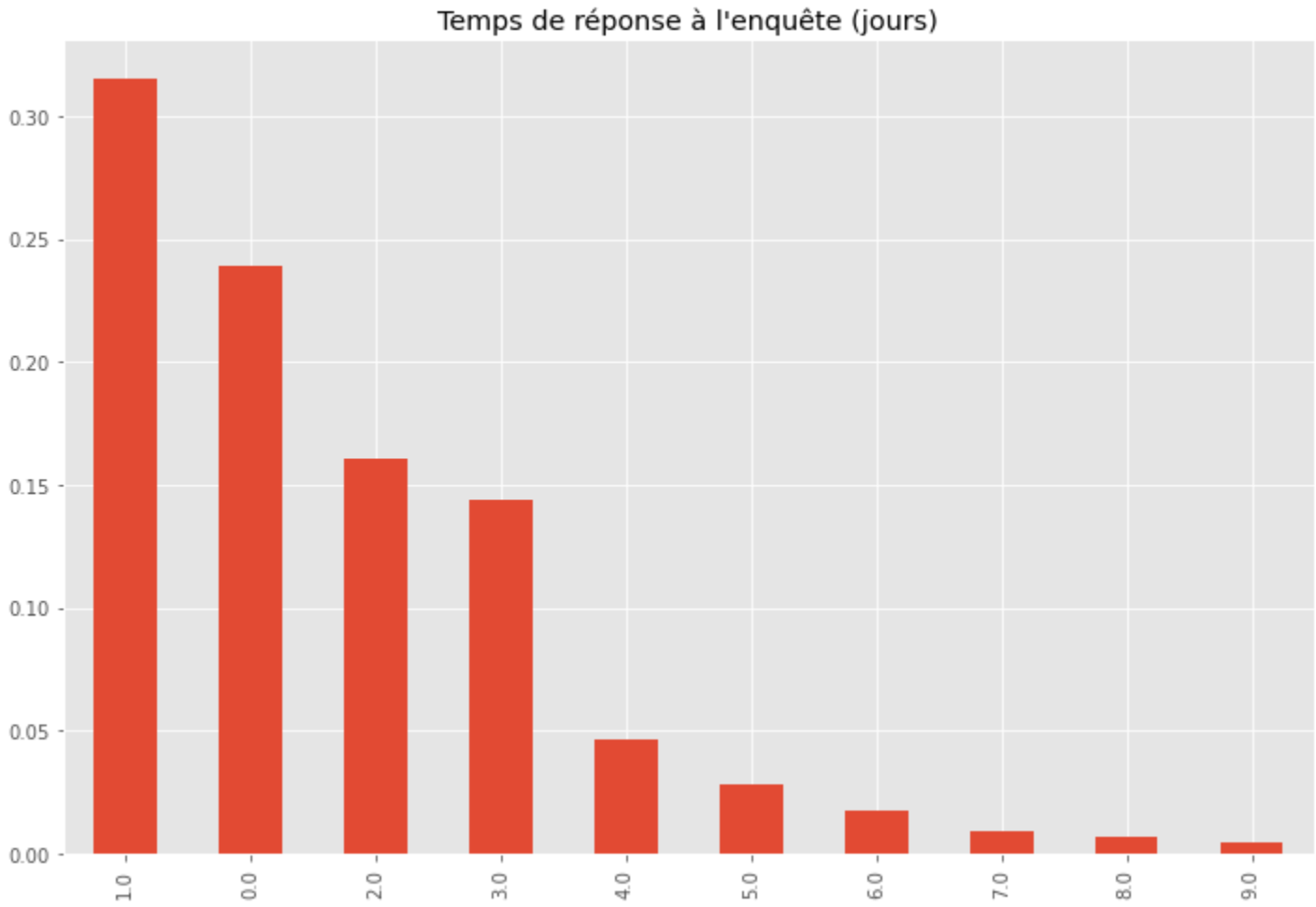
Les catégories de segmentation



Satisfaction des clients



Reactivité des clients



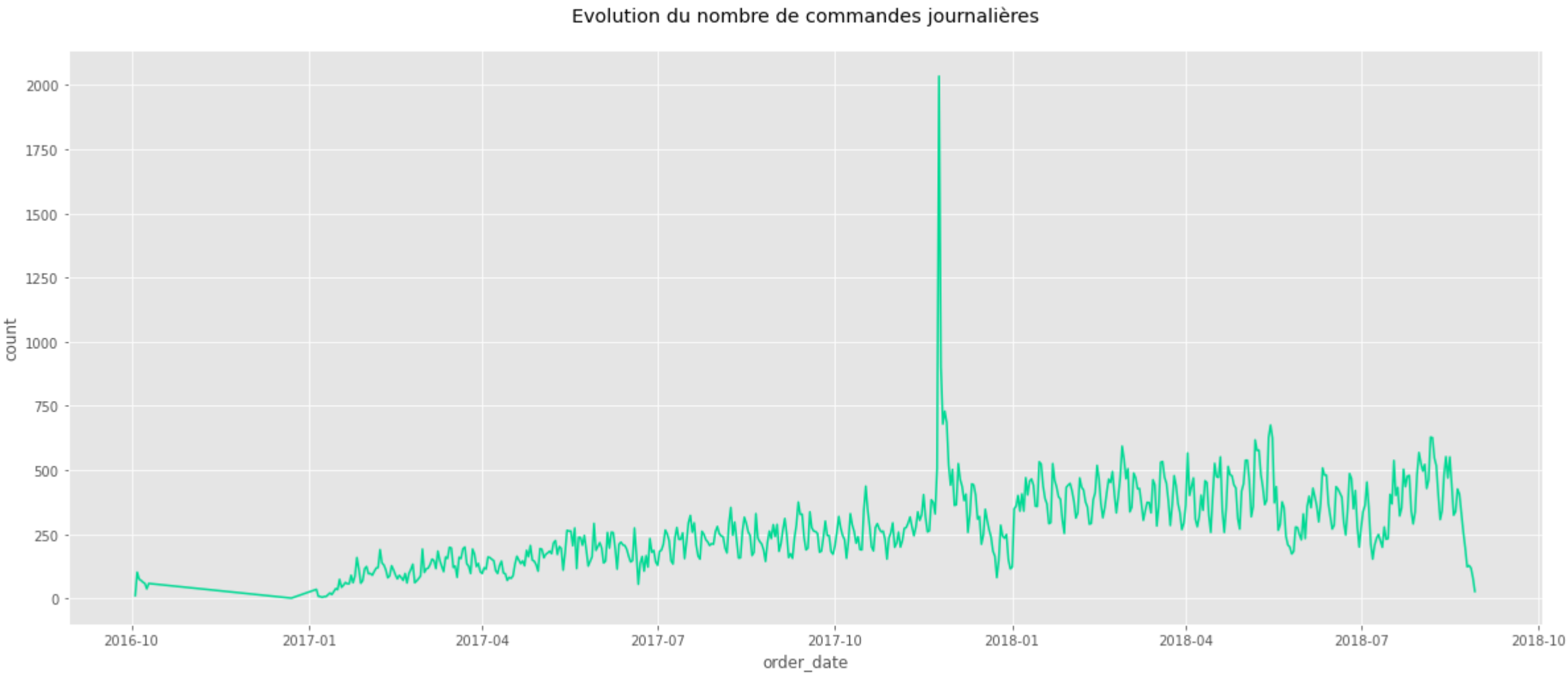
Describe de la dataframe

```
126]: data_numeric.describe()
```

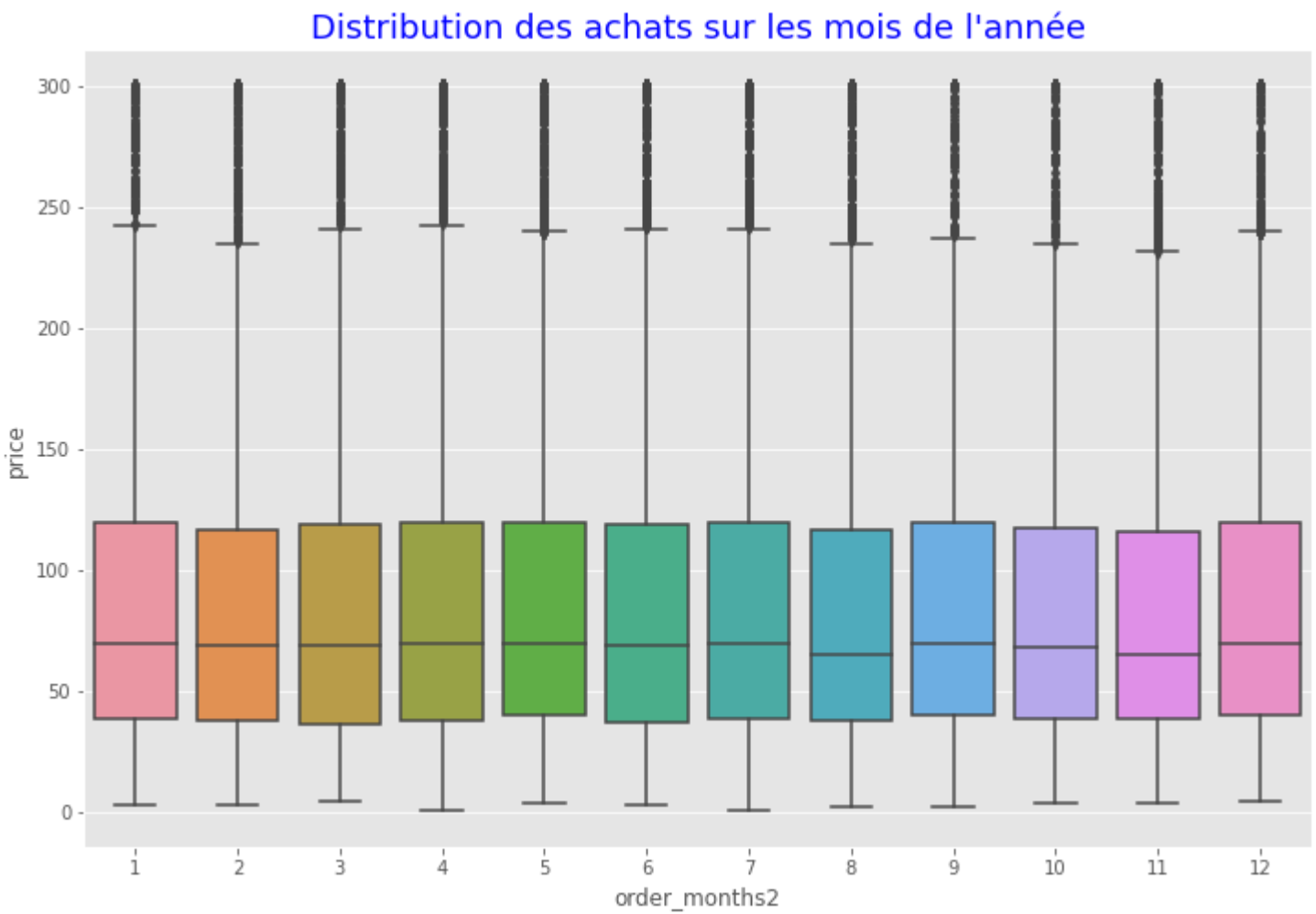
:

	price	freight_value	payment_sequential	payment_installments	payment_value	review_score	product_weight_g	delivered_time	revie
count	169358.000000	169358.000000	169358.000000	169358.000000	169358.000000	169358.000000	169358.000000	169358.000000	1693
mean	118.975550	19.573251	1.093630	2.914867	169.889156	4.089172	2091.787958	11.654720	
std	180.427053	15.438949	0.718767	2.765940	253.957683	1.340083	3750.961036	9.210226	
min	0.850000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	
25%	39.900000	12.790000	1.000000	1.000000	60.000000	4.000000	300.000000	6.000000	
50%	73.900000	16.110000	1.000000	1.000000	107.440000	5.000000	700.000000	9.000000	
75%	130.000000	20.850000	1.000000	4.000000	188.335000	5.000000	1800.000000	15.000000	
max	6735.000000	409.680000	26.000000	24.000000	13664.080000	5.000000	40425.000000	208.000000	5

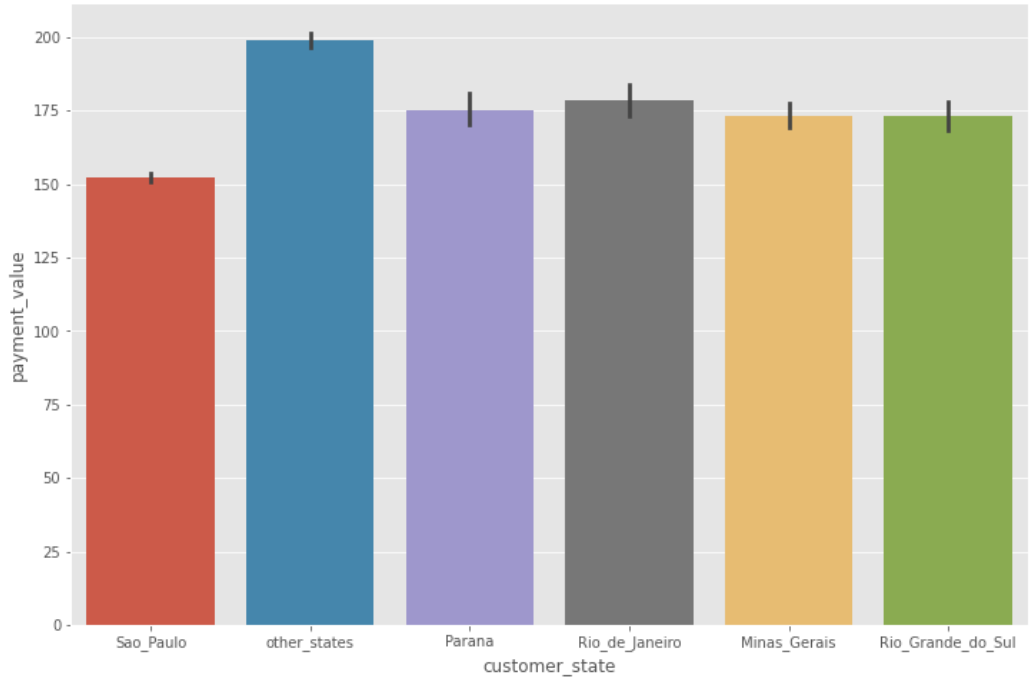
Evolution des commandes journalières

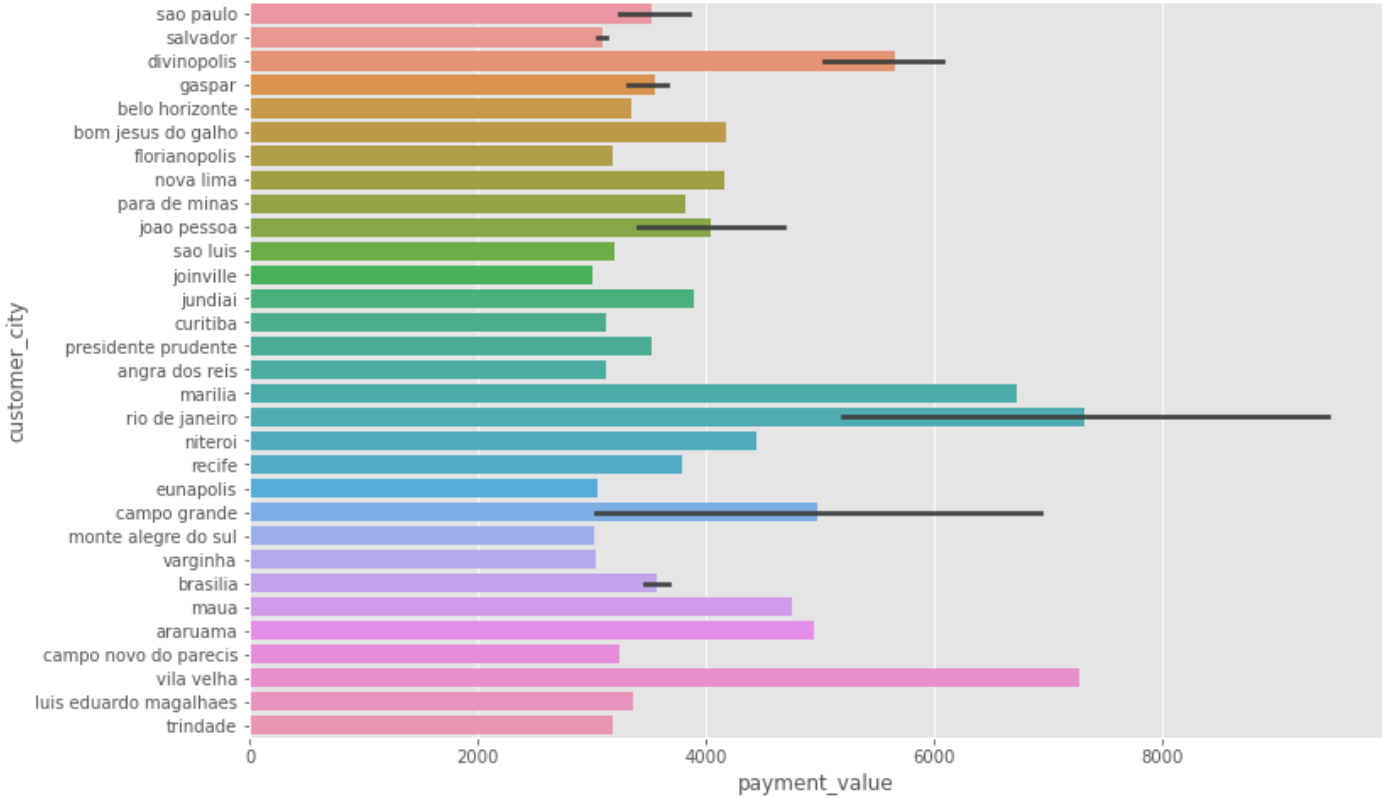


Les achats sur les mois de l’année

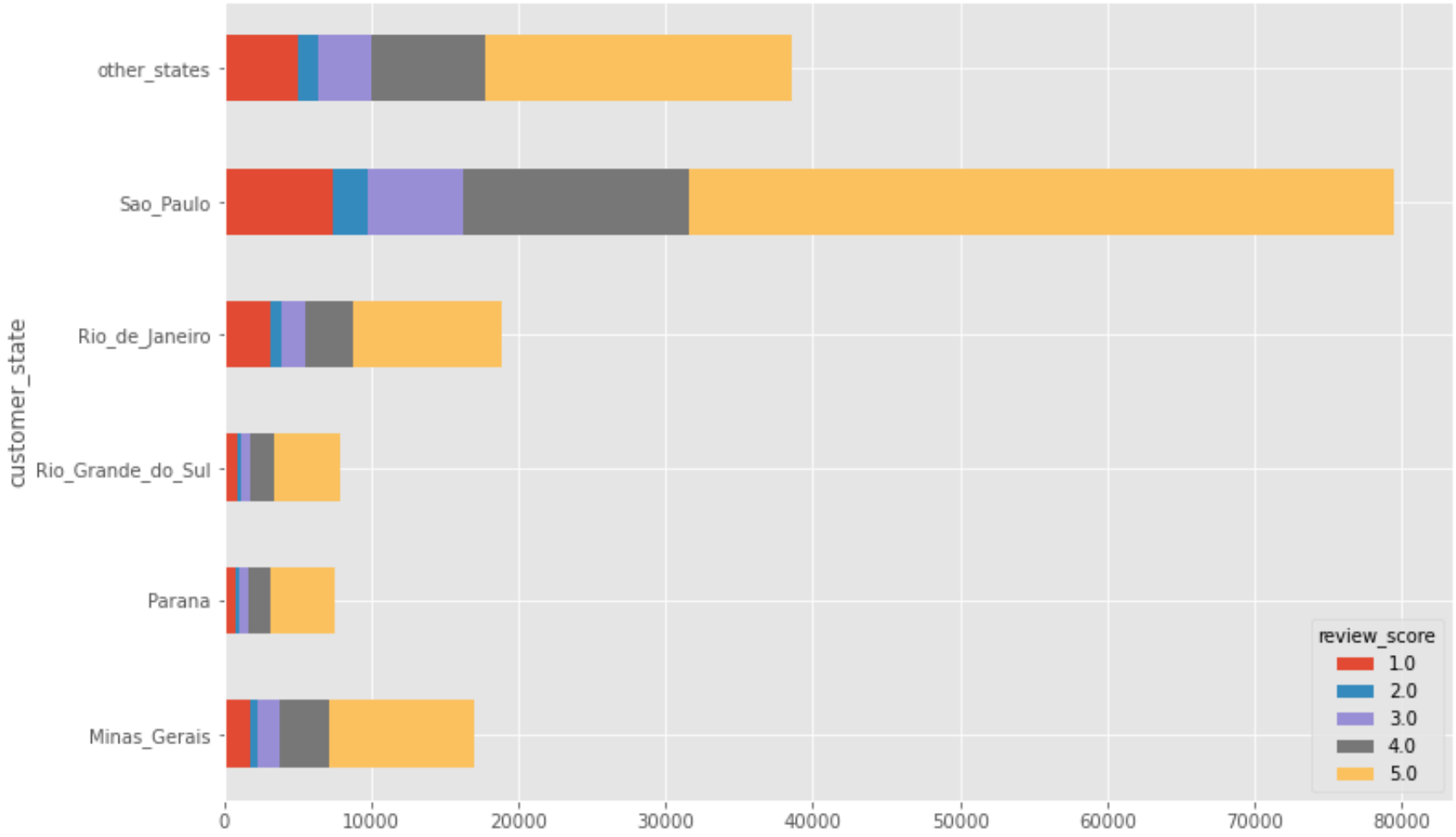


Ou se situent les meilleurs clients (payment > 3000) ?

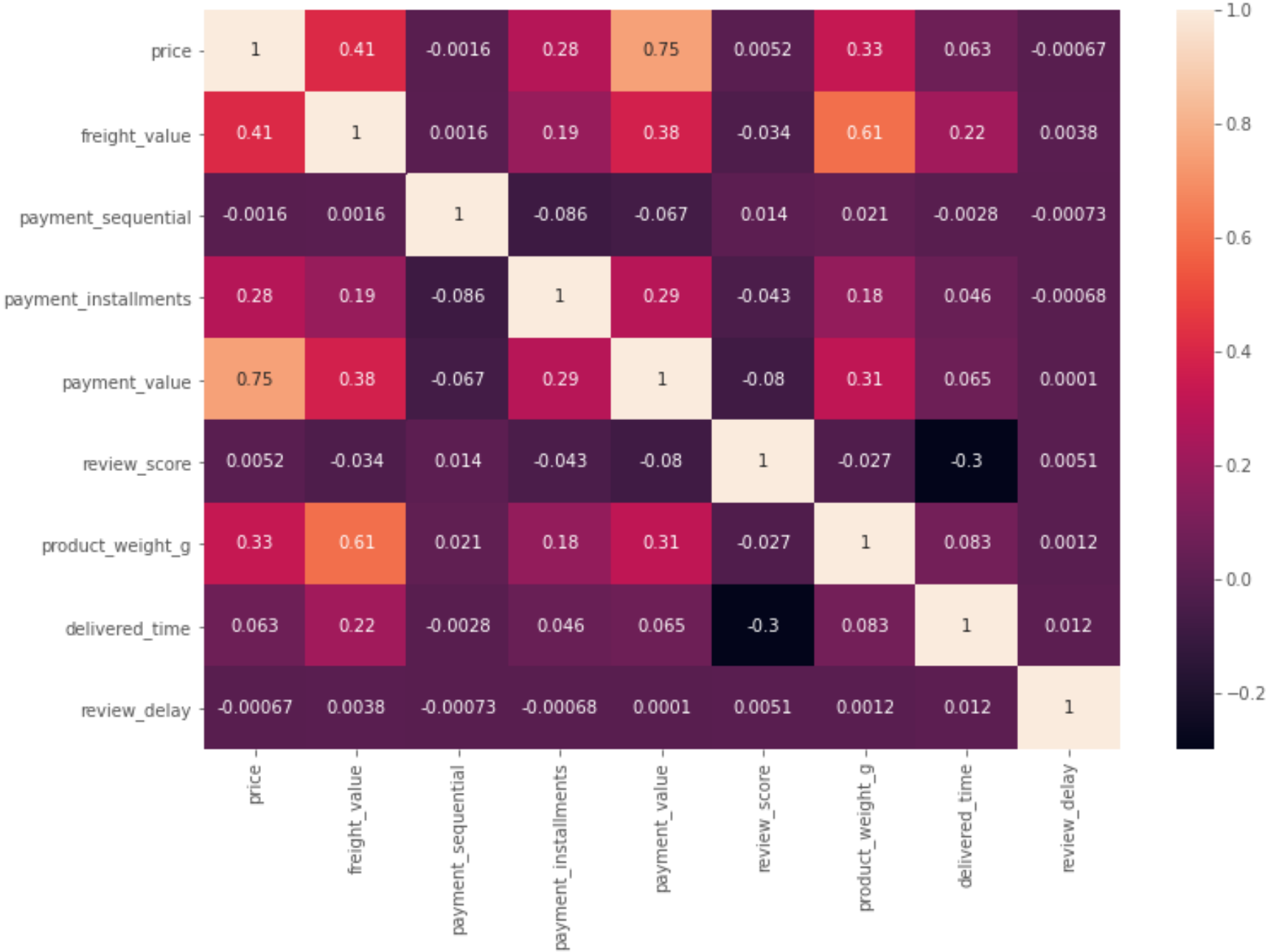




Satisfaction client et lieux de résidence



Heatmap


price 1 0.41 -0.0016 0.28 0.75 0.0052 0.33 0.063 -0.00067
freight_value 0.41 1 0.0016 0.19 0.38 -0.034 0.61 0.22 0.0038
payment_sequential -0.0016 0.0016 1 -0.086 -0.067 0.014 0.021 -0.0028 -0.00073
payment_installments 0.28 0.19 -0.086 1 0.29 -0.043 0.18 0.046 -0.00068
payment_value 0.75 0.38 -0.067 0.29 1 -0.08 0.31 0.065 0.0001
review_score 0.0052 -0.034 0.014 -0.043 -0.08 1 -0.027 -0.3 0.0051
product_weight_g 0.33 0.61 0.021 0.18 0.31 -0.027 1 0.083 0.0012
delivered_time 0.063 0.22 -0.0028 0.046 0.065 -0.3 0.083 1 0.012
review_delay -0.00067 0.0038 -0.00073 -0.00068 0.0001 0.0051 0.0012 0.012 1
price
freight_value
payment_sequential
payment_installments
payment_value
review_score
product_weight_g
delivered_time
review_delay
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

One hot encoding et standardisation

```
: numerical_features = list(encoding_data.select_dtypes(include=['int64','float64', 'uint8']).columns)
  numerical_features
```

```
['total_orders',
 'total_payment',
 'total_freight',
 'average_payment_sequential',
 'average_payment_installments',
 'average_review_score',
 'average_delivery_late',
 'average_delivery_time',
 'average_review_delay',
 'favorite_sale_month',
 'books_cds_media',
 'fashion_clothing_accessories',
 'flowers_gifts',
 'groceries_food_drink',
 'health_beauty',
 'home_furniture',
 'other',
 'sport',
 'technology',
 'toys_baby',
 'total_items',
 'customer_state_Minas_Gerais',
 'customer_state_Parana',
 'customer_state_Rio_Grande_do_Sul',
 'customer_state_Rio_de_Janeiro',
 'customer_state_Sao_Paulo',
 'customer_state_other_states']
```

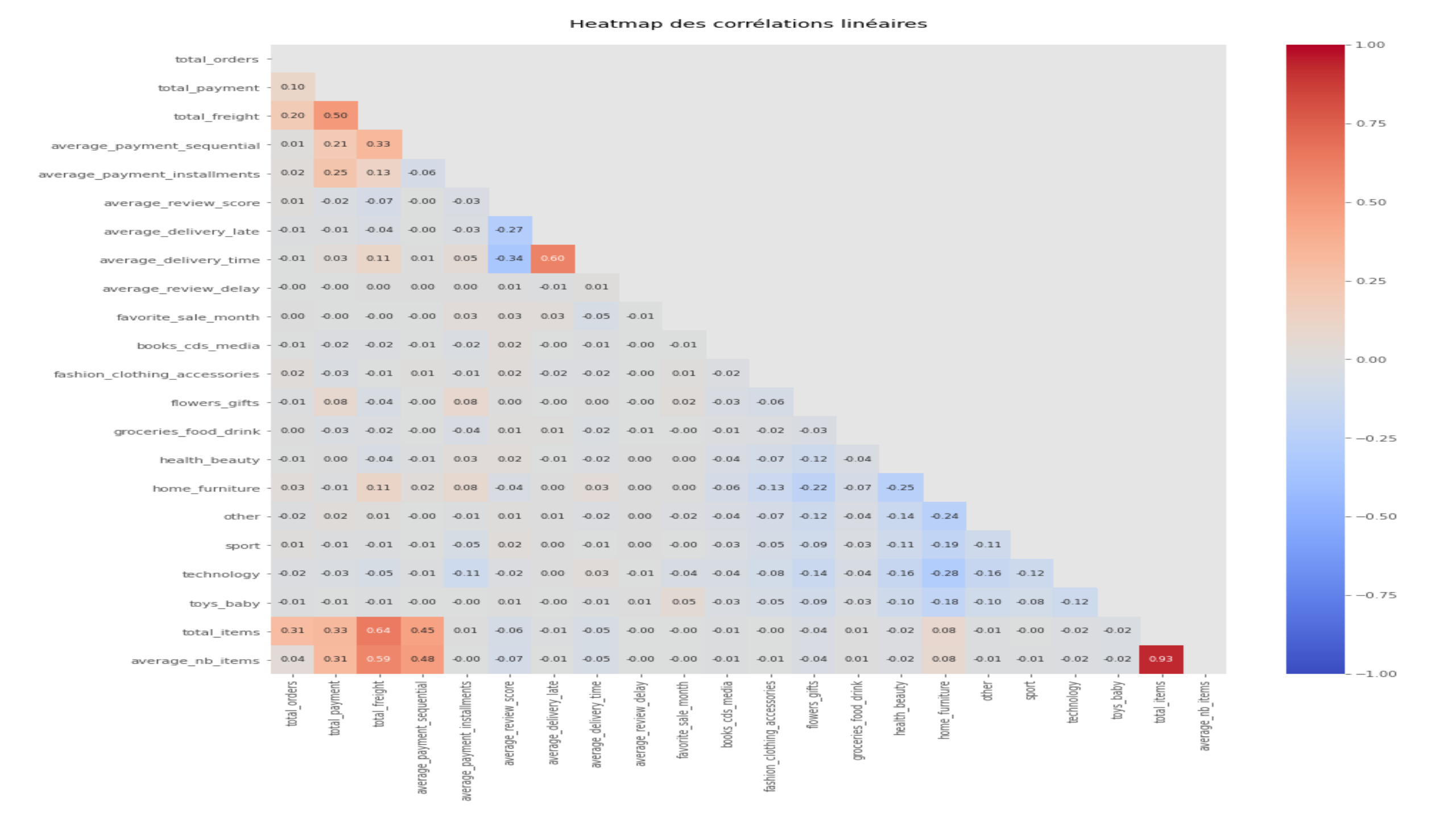
Aggregats clients

Groupons les données moyenne ou totales par client

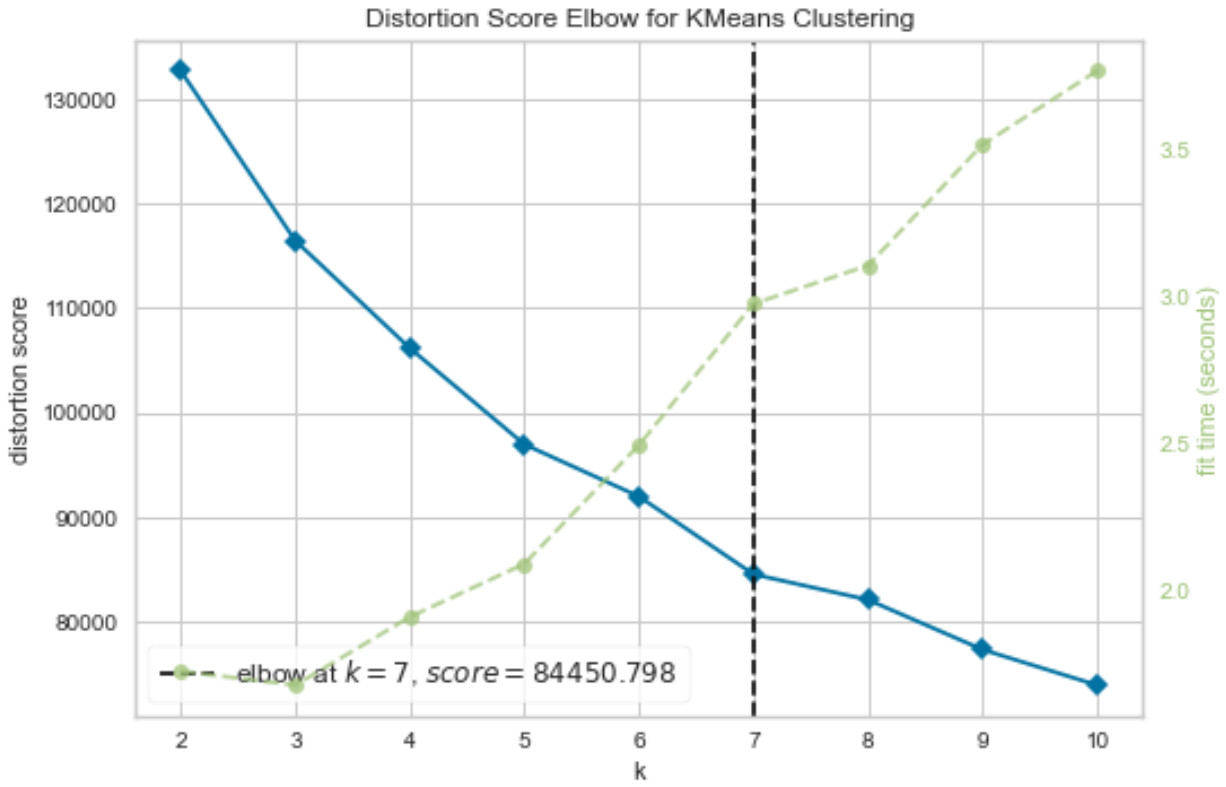
Nombre moyen d'articles par commande pour chaque client

Regroupement des consommateurs suivant les nouvelles catégories de produits

```
<class 'pandas.core.frame.DataFrame'>
Index: 91211 entries, 0000366f3b9a7992bf8c76cfdf3221e2 to f
Data columns (total 23 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   total_orders                    91211 non-null  int64
 1   total_payment                   91211 non-null  float64
 2   total_freight                   91211 non-null  float64
 3   average_payment_sequential      91211 non-null  float64
 4   average_payment_installments    91211 non-null  float64
 5   average_review_score            91211 non-null  float64
 6   average_delivery_late           91211 non-null  float64
 7   average_delivery_time           91211 non-null  float64
 8   average_review_delay            91211 non-null  float64
 9   favorite_sale_month             91211 non-null  int64
 10  books_cds_media                 91211 non-null  float64
 11  fashion_clothing_accessories    91211 non-null  float64
 12  flowers_gifts                   91211 non-null  float64
 13  groceries_food_drink            91211 non-null  float64
 14  health_beauty                   91211 non-null  float64
 15  home_furniture                  91211 non-null  float64
 16  other                           91211 non-null  float64
 17  sport                           91211 non-null  float64
 18  technology                      91211 non-null  float64
 19  toys_baby                       91211 non-null  float64
 20  total_items                     91211 non-null  float64
 21  average_nb_items                91211 non-null  float64
 22  customer_state                  91211 non-null  object
dtypes: float64(20), int64(2), object(1)
memory usage: 16.7+ MB
```


Heatmap des corrélations linéaires
total_orders
total_payment 0.10
total_freight 0.20 0.50
average_payment_sequential 0.01 0.21 0.33
average_payment_installments 0.02 0.25 0.13 -0.06
average_review_score 0.01 -0.02 -0.07 -0.00 -0.03
average_delivery_late -0.01 -0.01 -0.04 -0.00 -0.03 -0.27
average_delivery_time -0.01 0.03 0.11 0.01 0.05 -0.34 0.60
average_review_delay -0.00 -0.00 0.00 0.00 0.00 0.01 -0.01 0.01
favorite_sale_month 0.00 -0.00 -0.00 -0.00 0.03 0.03 0.03 -0.05 -0.01
books_cds_media -0.01 -0.02 -0.02 -0.01 -0.02 0.02 -0.00 -0.01 -0.00 -0.01
fashion_clothing_accessories 0.02 -0.03 -0.01 0.01 -0.01 0.02 -0.02 -0.02 -0.00 0.01 -0.02
flowers_gifts -0.01 0.08 -0.04 -0.00 0.08 0.00 -0.00 0.00 -0.00 0.02 -0.03 -0.06
groceries_food_drink 0.00 -0.03 -0.02 -0.00 -0.04 0.01 0.01 -0.02 -0.01 -0.00 -0.01 -0.02 -0.03
health_beauty -0.01 0.00 -0.04 -0.01 0.03 0.02 -0.01 -0.02 0.00 0.00 -0.04 -0.07 -0.12 -0.04
home_furniture 0.03 -0.01 0.11 0.02 0.08 -0.04 0.00 0.03 0.00 0.00 -0.06 -0.13 -0.22 -0.07 -0.25
other -0.02 0.02 0.01 -0.00 -0.01 0.01 0.01 -0.02 0.00 -0.02 -0.04 -0.07 -0.12 -0.04 -0.14 -0.24
sport 0.01 -0.01 -0.01 -0.01 -0.05 0.02 0.00 -0.01 0.00 -0.00 -0.03 -0.05 -0.09 -0.03 -0.11 -0.19 -0.11
technology -0.02 -0.03 -0.05 -0.01 -0.11 -0.02 0.00 0.03 -0.01 -0.04 -0.04 -0.08 -0.14 -0.04 -0.16 -0.28 -0.16 -0.12
toys_baby -0.01 -0.01 -0.01 -0.00 -0.00 0.01 -0.00 -0.01 0.01 0.05 -0.03 -0.05 -0.09 -0.03 -0.10 -0.18 -0.10 -0.08 -0.12
total_items 0.31 0.33 0.64 0.45 0.01 -0.06 -0.01 -0.05 -0.00 -0.00 -0.01 -0.00 -0.04 0.01 -0.02 0.08 -0.01 -0.00 -0.02 -0.02
average_nb_items 0.04 0.31 0.59 0.48 -0.00 -0.07 -0.01 -0.05 -0.00 -0.00 -0.01 -0.01 -0.04 0.01 -0.02 0.08 -0.01 -0.01 -0.02 -0.02 0.93
total_orders
total_payment
total_freight
average_payment_sequential
average_payment_installments
average_review_score
average_delivery_late
average_delivery_time
average_review_delay
favorite_sale_month
books_cds_media
fashion_clothing_accessories
flowers_gifts
groceries_food_drink
health_beauty
home_furniture
other
sport
technology
toys_baby
total_items
average_nb_items
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

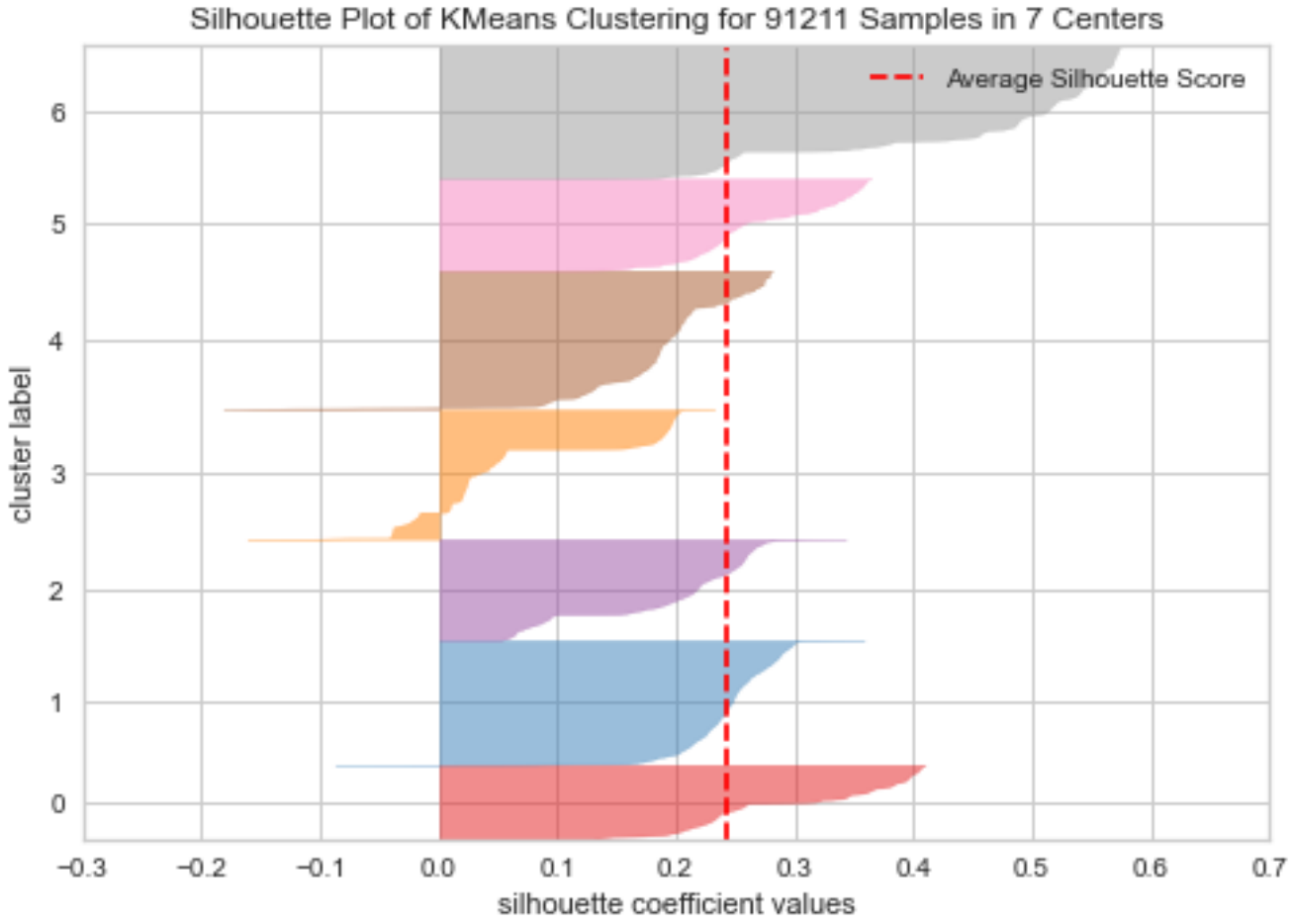
Clustering Kmeans : méthode du coude pour déterminer le meilleur K



le point du coude est celui du nombre de clusters à partir duquel la variance ne se réduit plus significativement

K = 7

Coefficients de silhouettes complémentaire pour déterminer le K optimal



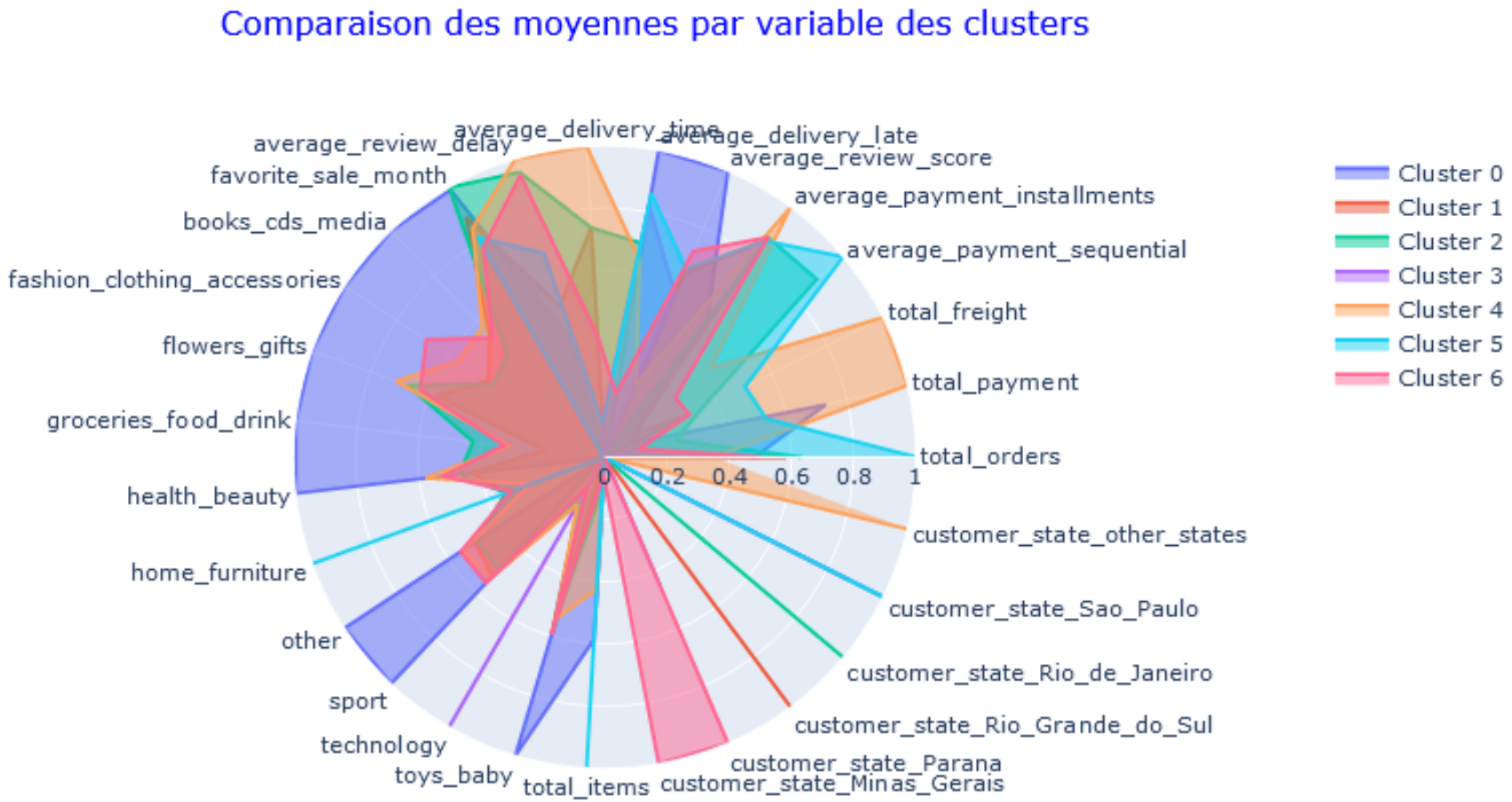
Ici un K optimal serait 7 d'après les coefficients de silhouette

Les 7 premiers clusters suivant leurs moyennes

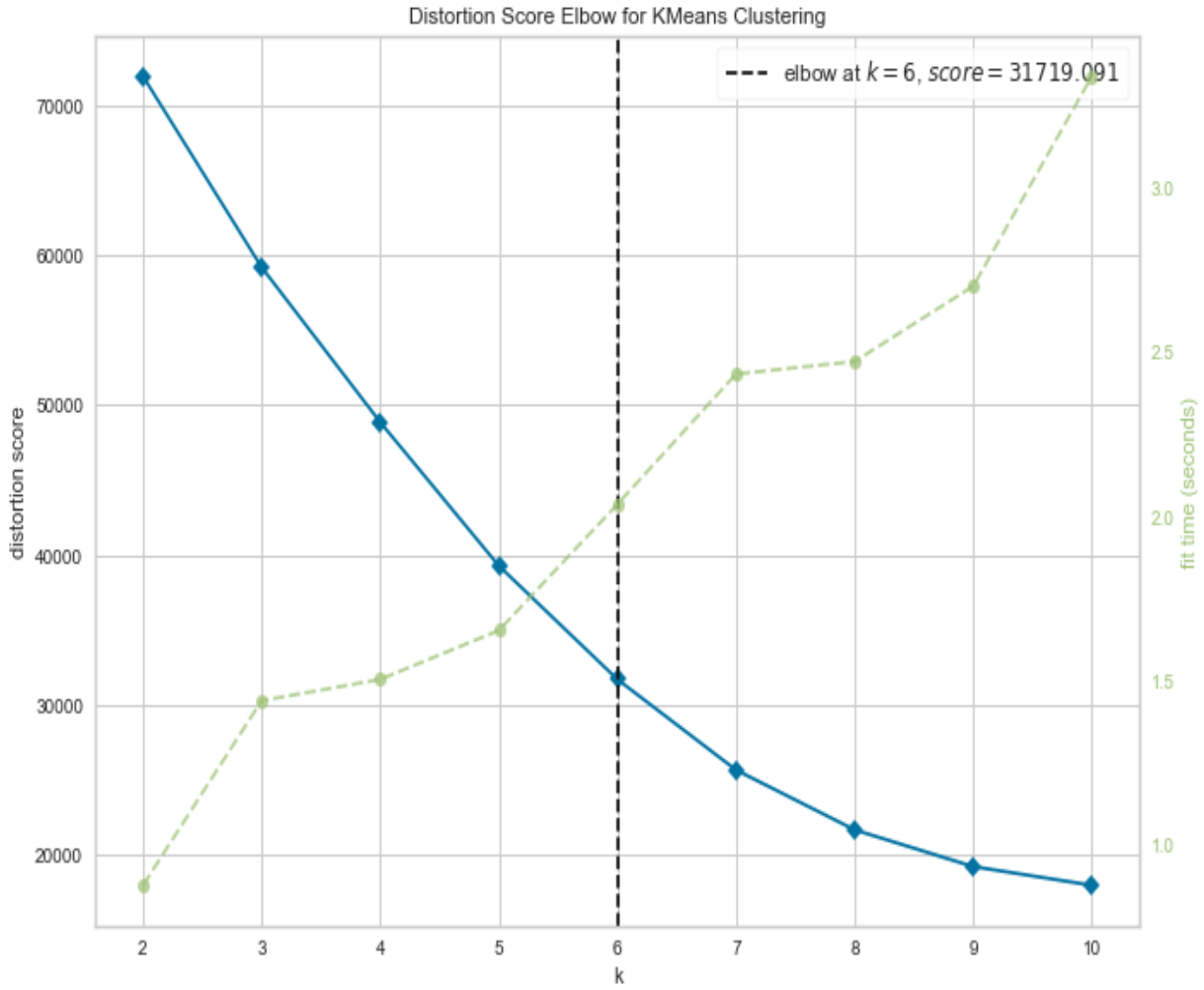
kmeans_label	total_orders	total_payment	total_freight	average_payment_sequential	average_payment_installments	average_review_score	average_
0	0.002462	0.010813	0.010105	0.001566	0.110056	0.825918	
1	0.002642	0.008454	0.011926	0.001537	0.125845	0.795626	
2	0.002732	0.009214	0.012250	0.002234	0.126405	0.740430	
3	0.001616	0.008833	0.009275	0.001437	0.082448	0.789203	
4	0.002232	0.011685	0.017798	0.001835	0.132391	0.762177	
5	0.003378	0.010179	0.013574	0.002334	0.125682	0.796730	
6	0.002496	0.008815	0.011873	0.001703	0.126383	0.801990	

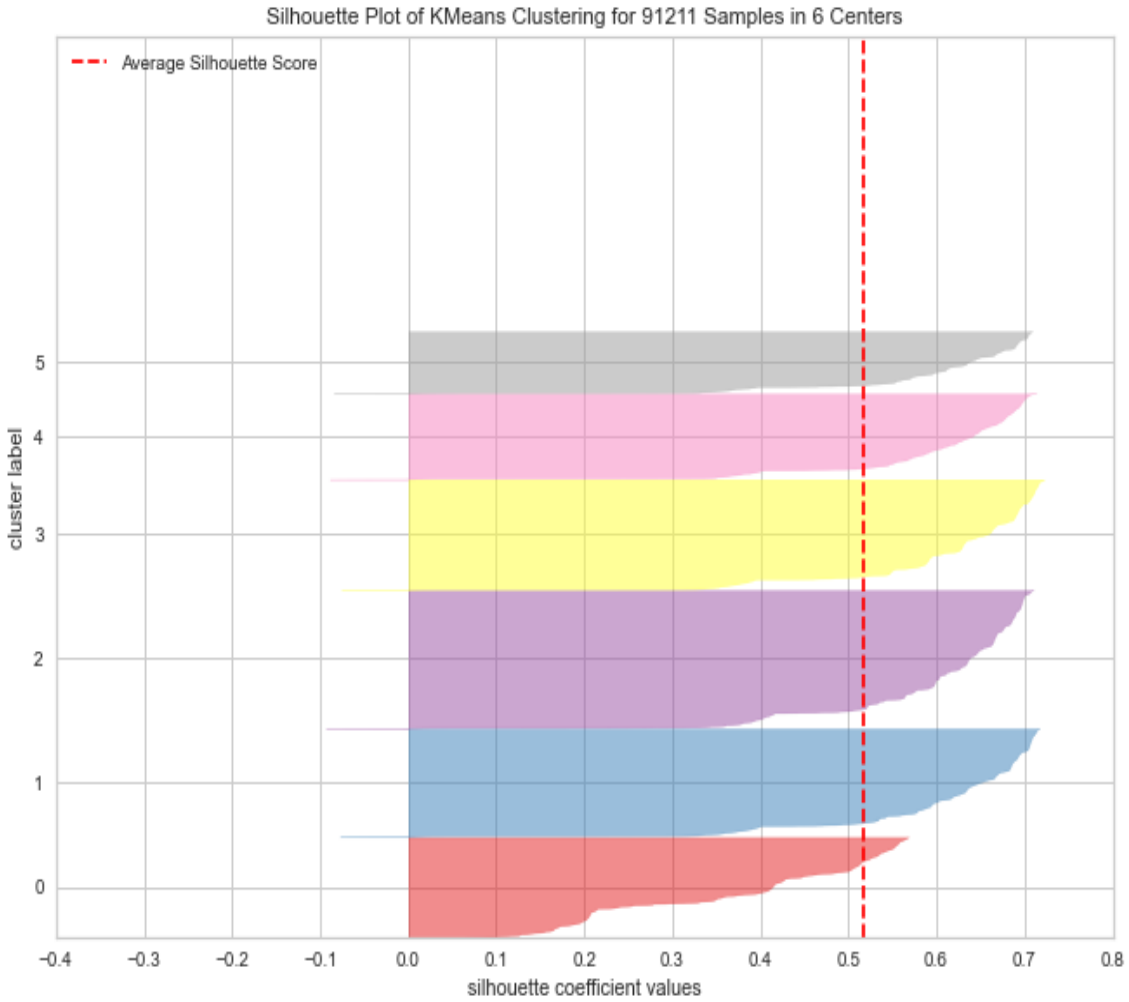
7 rows × 27 columns

Radar chart des clusters projetés



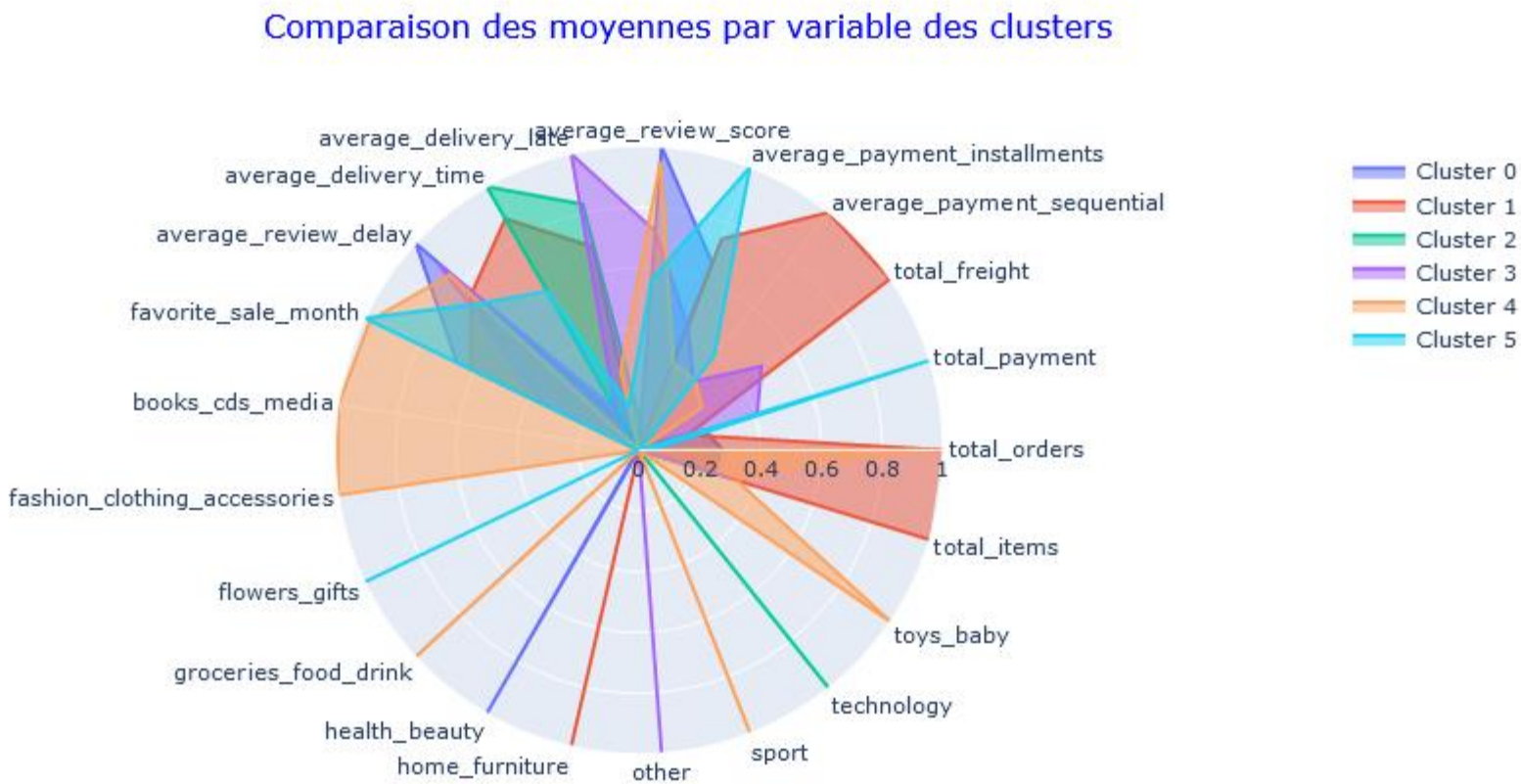
Nouveau Kmeans sans les variables Etats





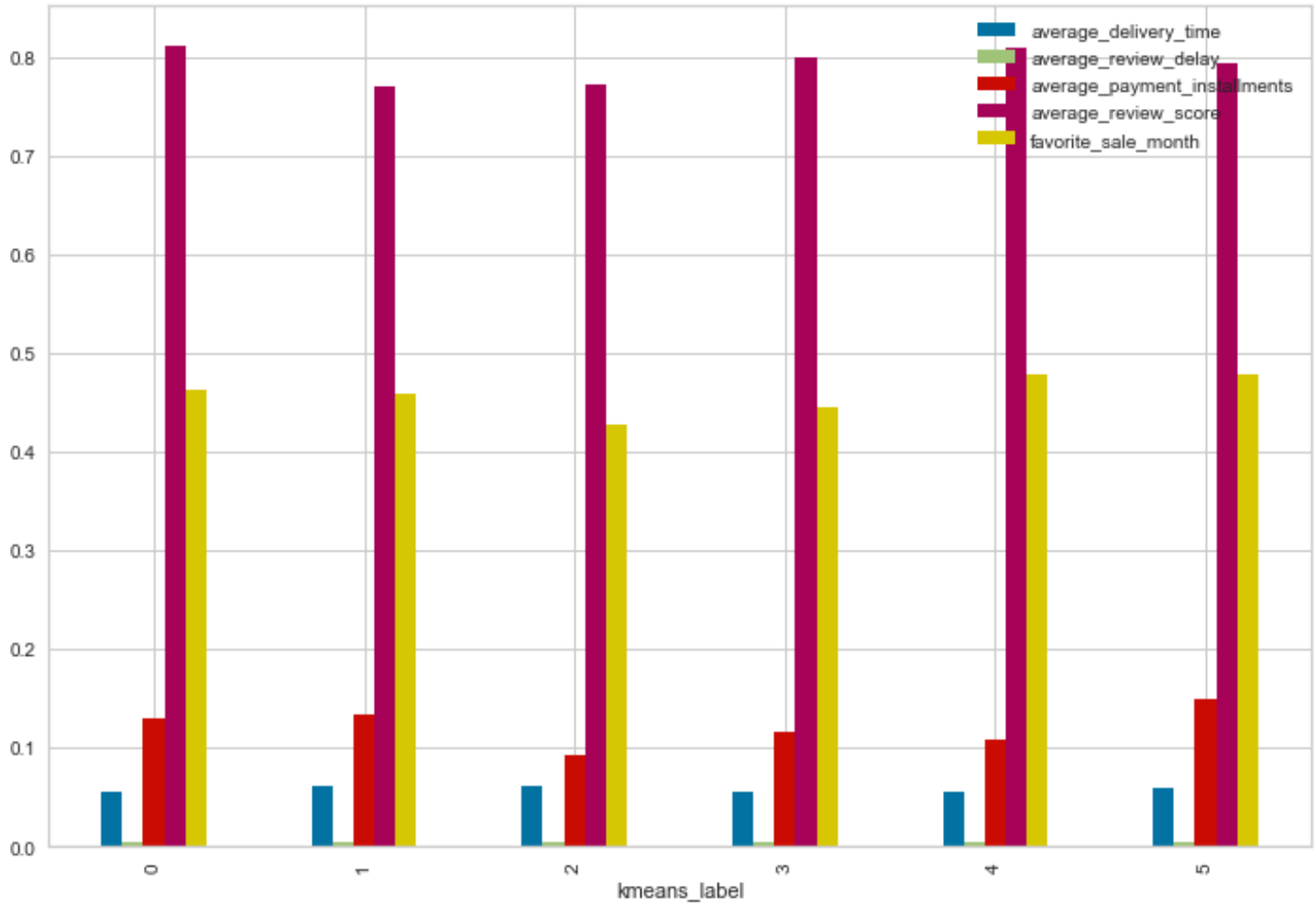
K vaut 6 et le score silhouette moyen est passé à 0.62

Nouveau Radar chart projeté

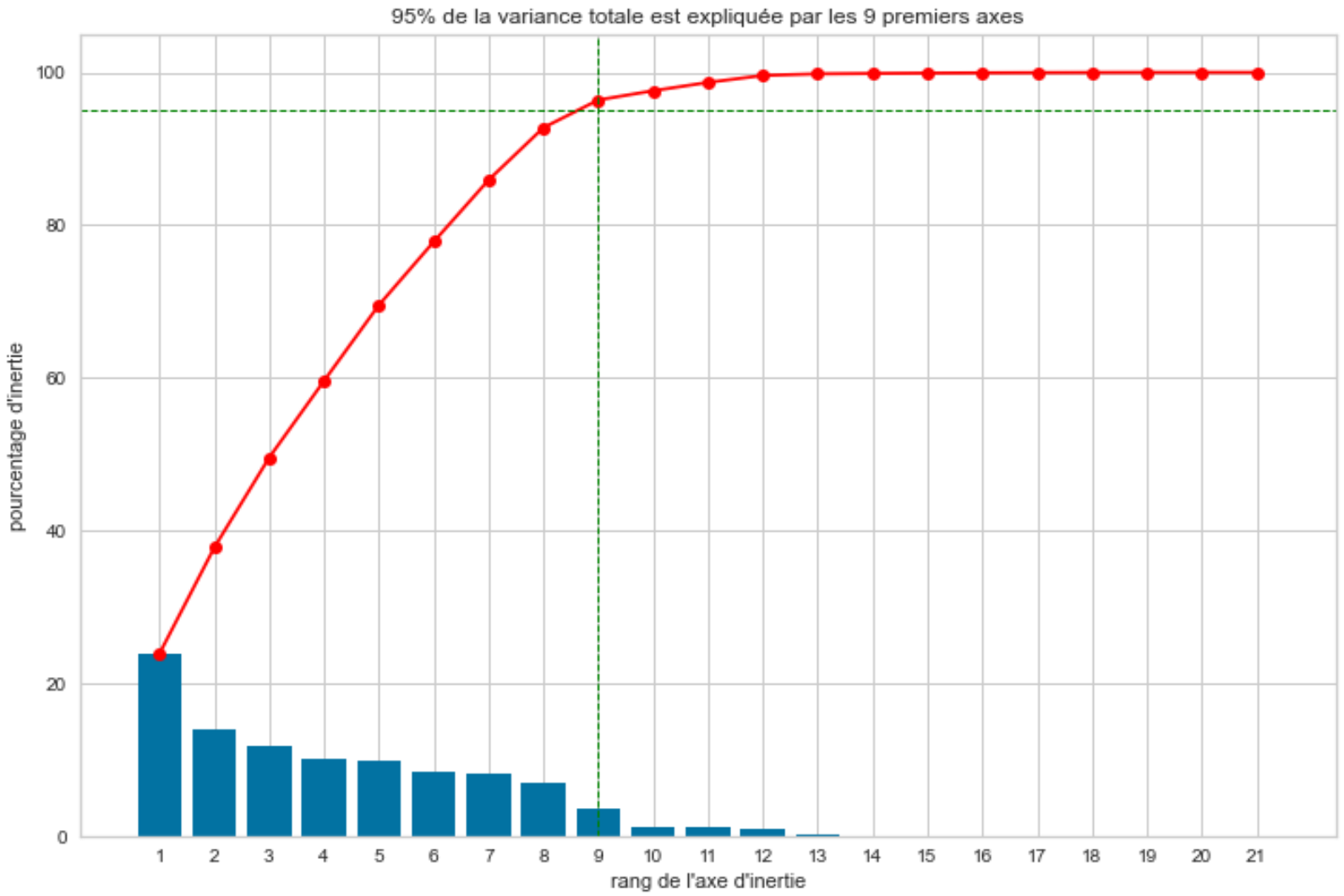


Clustering par rapport à des variables de sélection

groupe 0 : clients très satisfaits - bon délai livraison - bcp paiements échelonnés
groupe 1 : clients mécontents - délais livraison élevés - bcp paiements échelonnés
groupe 2 : peu de paiements échelonnés) - la plus basse saison mensuelle d'achat
groupe 3 : clients plutôt satisfaits - livraison moyenne - achète dans une autre période de l'année
groupe 4 : clients plutôt satisfaits - paiements échelonnés moyens - mois forts d'achat - livraison moyenne
groupe 5 : bcp paiements échelonnés - mois forts d'achat - assez satisfaits - livraison longue

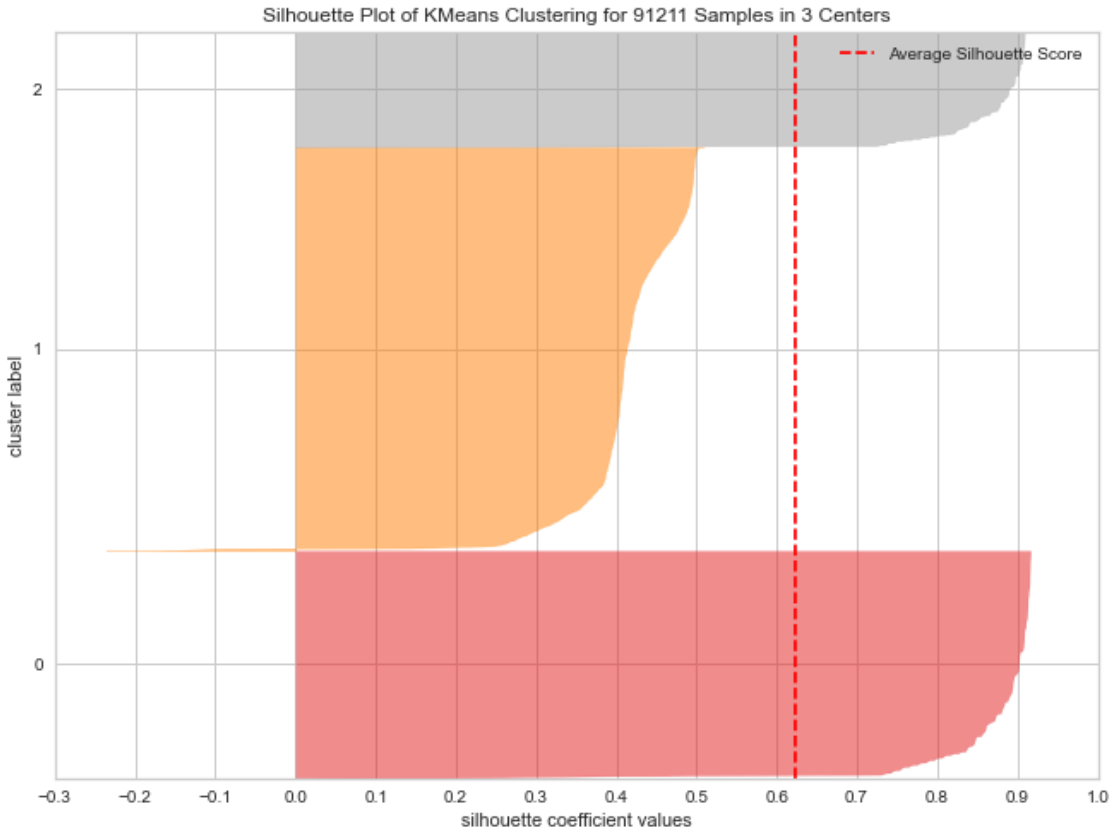


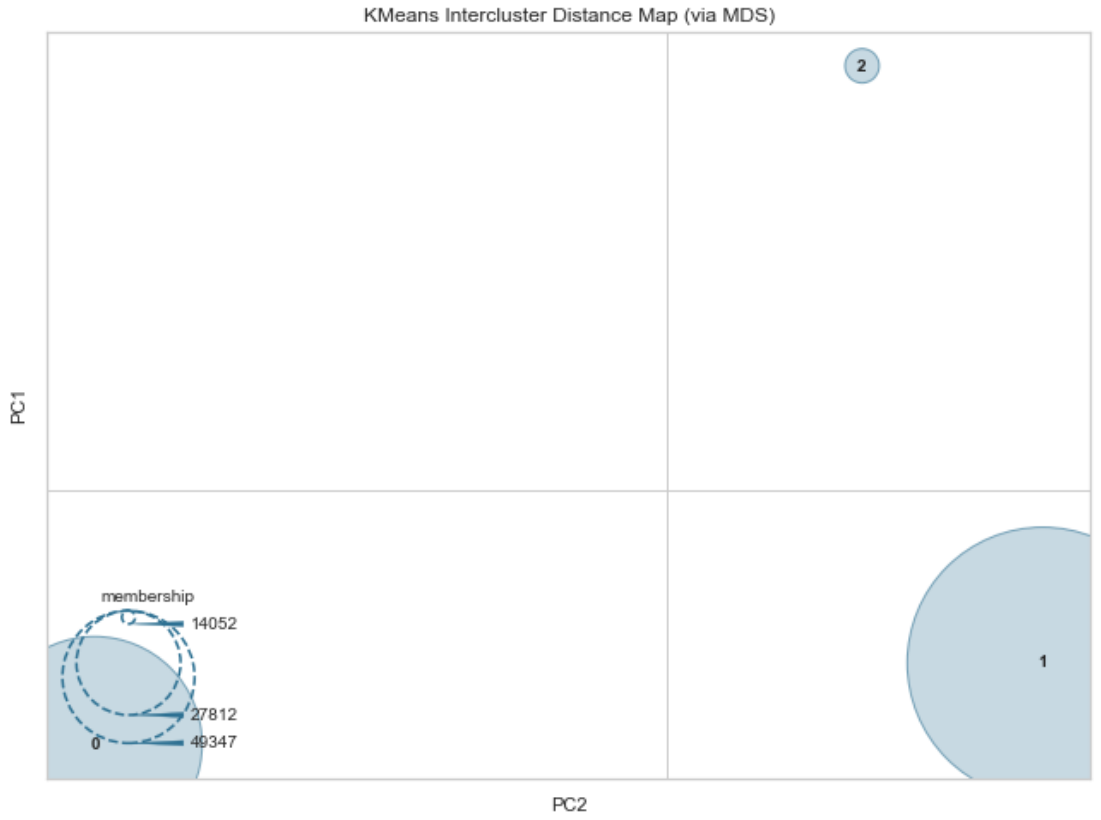
Reduction de dimension avec PCA



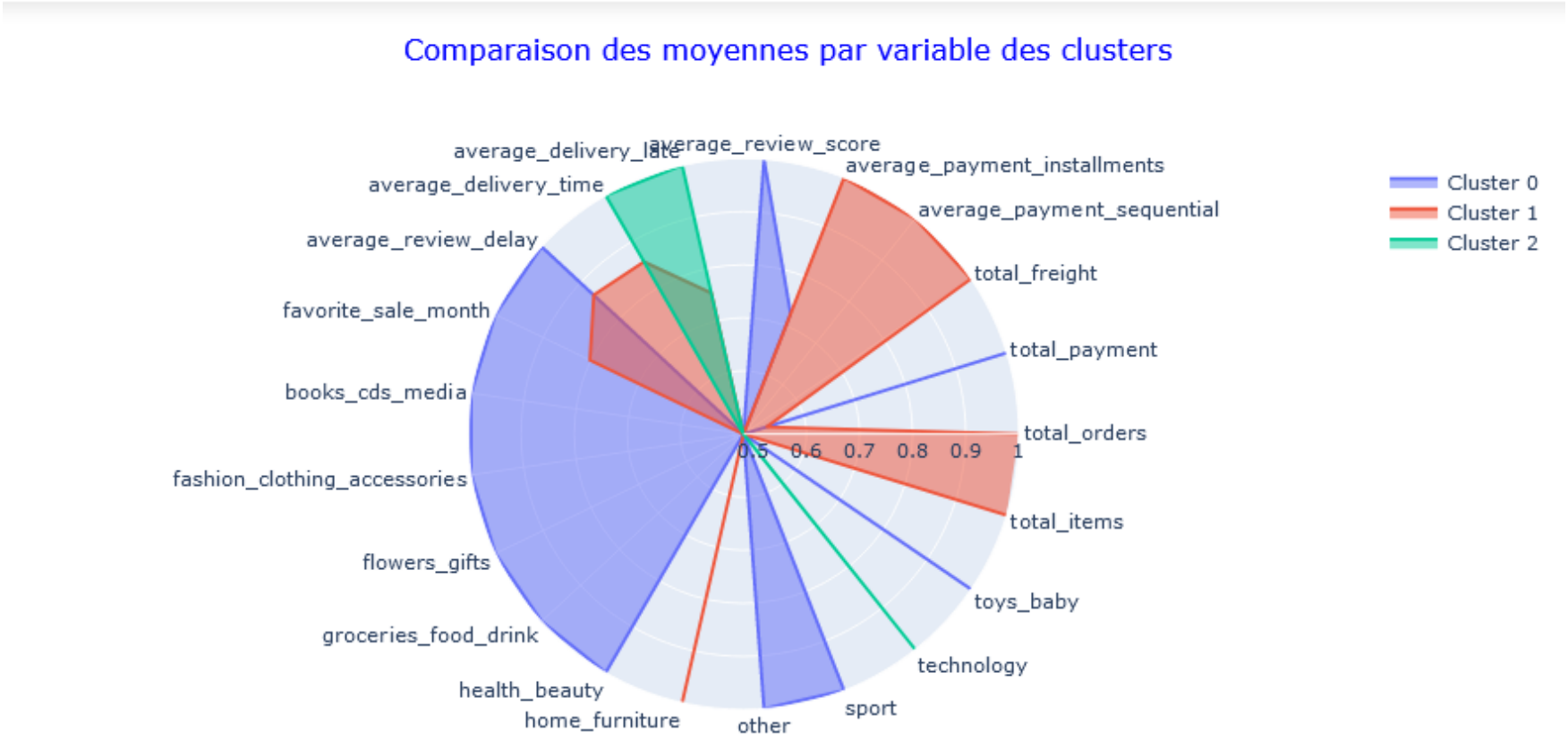
Il faut donc conserver 9 axes principaux pour expliquer la variance à 95%.

Silhouettes score et distances inter clusters





Radar chart sur les variables projetées



La PCA améliore peu la qualité du modèle (silhouette score reste à 0.62)

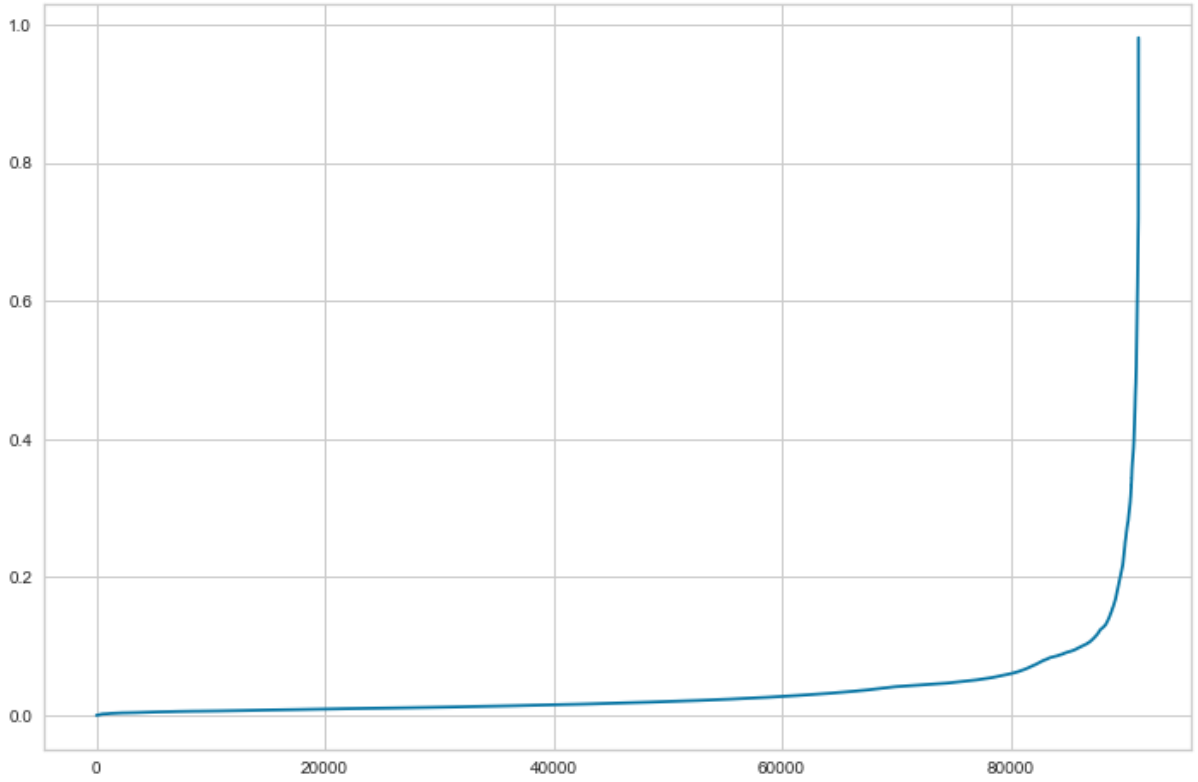
Scores de stabilité des kmeans à l’initialisation

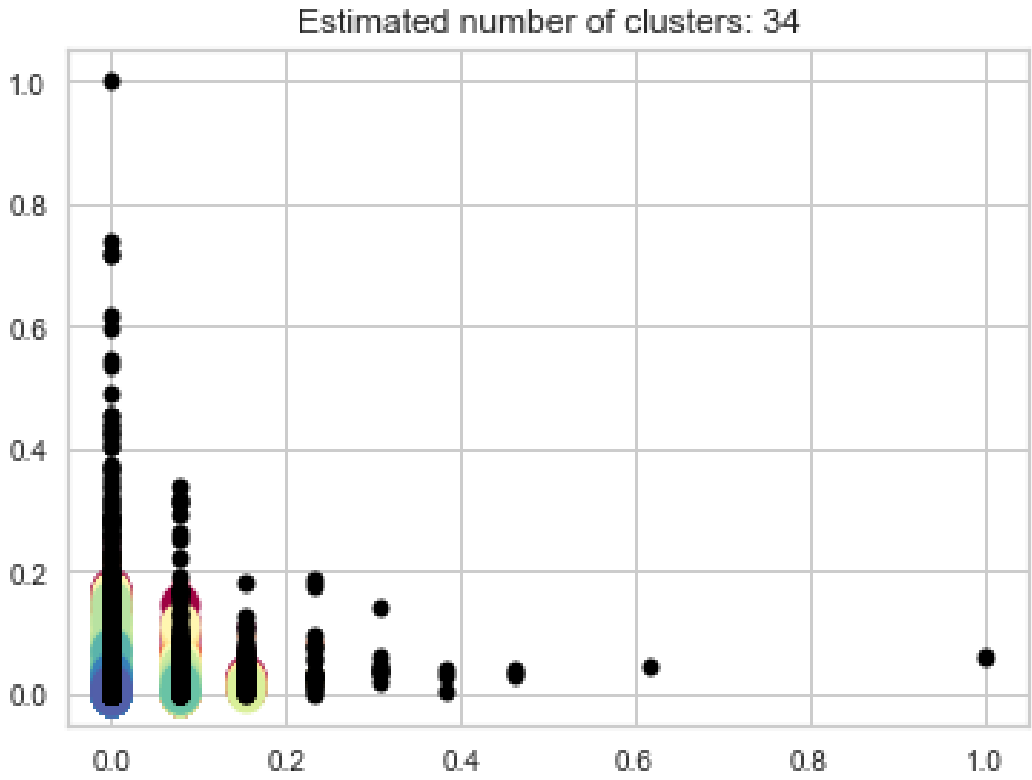
Scores de stabilité à l'initialisation

Iteration	FitTime	Inertia	Homo	ARI	AMI
Iter 0	0.749s	15154	0.616	0.671	0.758
Iter 1	0.109s	15154	0.616	0.671	0.758
Iter 2	0.131s	15154	0.616	0.671	0.758
Iter 3	0.091s	21465	0.429	0.390	0.597
Iter 4	0.101s	15154	0.616	0.671	0.758
Iter 5	0.121s	15154	0.616	0.671	0.758
Iter 6	0.093s	21465	0.429	0.390	0.597
Iter 7	0.142s	15154	0.616	0.671	0.758
Iter 8	0.101s	15154	0.616	0.671	0.758
Iter 9	0.111s	15154	0.616	0.671	0.758
Iter 10	0.093s	21465	0.429	0.390	0.597
Iter 11	0.123s	15154	0.616	0.671	0.758
Iter 12	0.123s	15154	0.616	0.671	0.758
Iter 13	0.131s	15154	0.616	0.671	0.758
Iter 14	0.131s	15154	0.616	0.671	0.758

Les différentes itérations montrent des inerties proches, une bonne homogénéité et un score AMI acceptable.
La stabilité à l'initialisation du modèle K-Means est bonne

DBSCAN : choix de Epsilon





Estimated number of clusters: 34
Estimated number of noise points: 33435
Silhouette Coefficient: 0.144

DBSCAN semble pas adapté pour notre échantillon
Les données sont trop éparpillées

Stabilité du clustering dans le temps

La période complète des commandes porte sur 23 mois : 2016-10-03 au 2017-10-03

Principe : Recalculer toutes les features en fonction d'une période initiale donnée (12 mois depuis la plus petite date de commande)
Ensuite rajouter successivement (10 fois) des périodes de 1 mois

Clustering sans les catégories produits et les Etats des clients

'kmeans_init' est le modèle de base issu de la clusterisation de la Baseline

Pour calculer la stabilité des clusters, on calcule pour chaque base périodique , le score ARI entre la prédiction de la database périodique sur le modèle initiale 'kmeans_init' et le label issu du nouveau modèle issu de la clusterisation de la database périodique.

Baseline : 12 mois depuis la plus petite période de commande

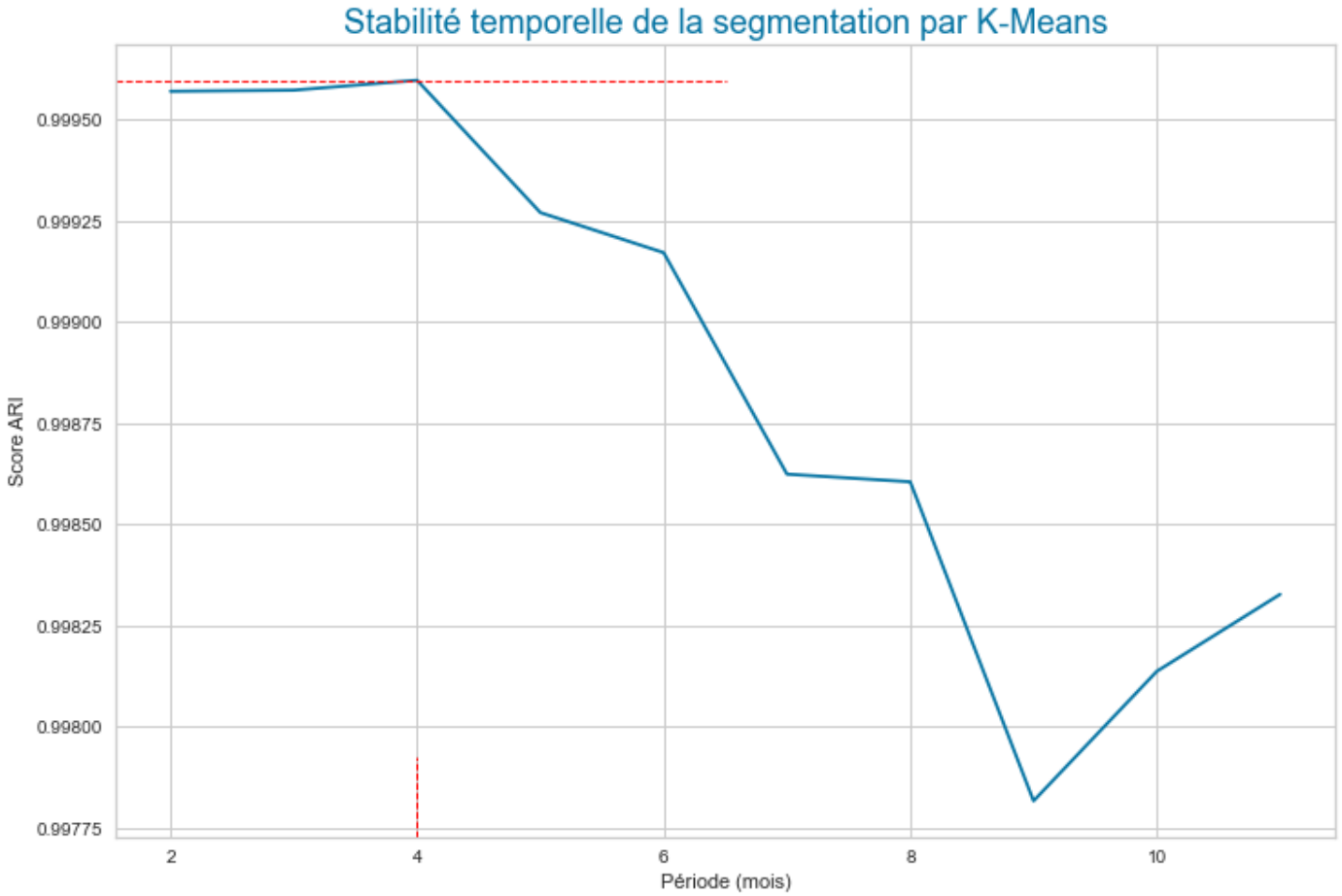
```
<class 'pandas.core.frame.DataFrame'>
Index: 25230 entries, 0000f46a3911fa3c0805444483337064 to ffffd2657e2aad2907e67c3e9daecbeb
Data columns (total 28 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   total_orders                        25230 non-null  int64
 1   total_payment                       25230 non-null  float64
 2   total_freight                       25230 non-null  float64
 3   average_payment_sequential          25230 non-null  float64
 4   average_payment_installments        25230 non-null  float64
 5   average_review_score                25230 non-null  float64
 6   average_delivery_late               25230 non-null  float64
 7   average_delivery_time               25230 non-null  float64
 8   average_review_delay                25230 non-null  float64
 9   favorite_sale_month                 25230 non-null  int64
 10  books_cds_media                     25230 non-null  float64
 11  fashion_clothing_accessories        25230 non-null  float64
 12  flowers_gifts                       25230 non-null  float64
 13  groceries_food_drink                25230 non-null  float64
 14  health_beauty                       25230 non-null  float64
 15  home_furniture                      25230 non-null  float64
 16  other                               25230 non-null  float64
 17  sport                               25230 non-null  float64
 18  technology                          25230 non-null  float64
 19  toys_baby                           25230 non-null  float64
 20  total_items                         25230 non-null  float64
 21  mean_nb_items                       25230 non-null  float64
 22  customer_state_Minas_Gerais         25230 non-null  uint8
 23  customer_state_Parana               25230 non-null  uint8
 24  customer_state_Rio_Grande_do_Sul    25230 non-null  uint8
 25  customer_state_Rio_de_Janeiro       25230 non-null  uint8
 26  customer_state_Sao_Paulo            25230 non-null  uint8
 27  customer_state_other_states         25230 non-null  uint8
dtypes: float64(20), int64(2), uint8(6)
memory usage: 4.6+ MB
```

Kmeans de la Baseline : 6 clusters par leurs moyennes

kmeans_label_init	total_orders	total_payment	total_freight	average_payment_sequential	average_payment_installments	average_review_score	average_delivery_
0	0.006910	0.008443	0.011885	0.001899	0.097049	0.836497	0.41
1	0.007359	0.009537	0.014528	0.003062	0.124725	0.787445	0.41
2	0.002824	0.010313	0.010918	0.001929	0.081423	0.798009	0.41
3	0.005763	0.010419	0.011015	0.001895	0.121908	0.813362	0.40
4	0.002669	0.014481	0.011070	0.002185	0.136371	0.831127	0.41
5	0.004806	0.011666	0.012764	0.002285	0.115240	0.809190	0.41

6 rows × 22 columns

Contrat de maintenance



Sur ce plot des scores ARI par périodes glissantes de 1 mois , plateau après 4 mois puis la courbe décroit jusqu'à la période 9 mois. Elle change d'allure (croissance sur 4 mois) après 9 périodes glissantes de 1 mois.
Prévoir la maintenance du programme de segmentation tous les 9 mois dans un premier temps,
puis re-tester cette stabilité temporelle au fil du temps afin de l'affiner.
Les segments clients seront redéfinis à chaque maintenance. 'average_review_score' et 'favorite_sale_month' sont des variables qui peuvent avoir un impact sur les clusters.