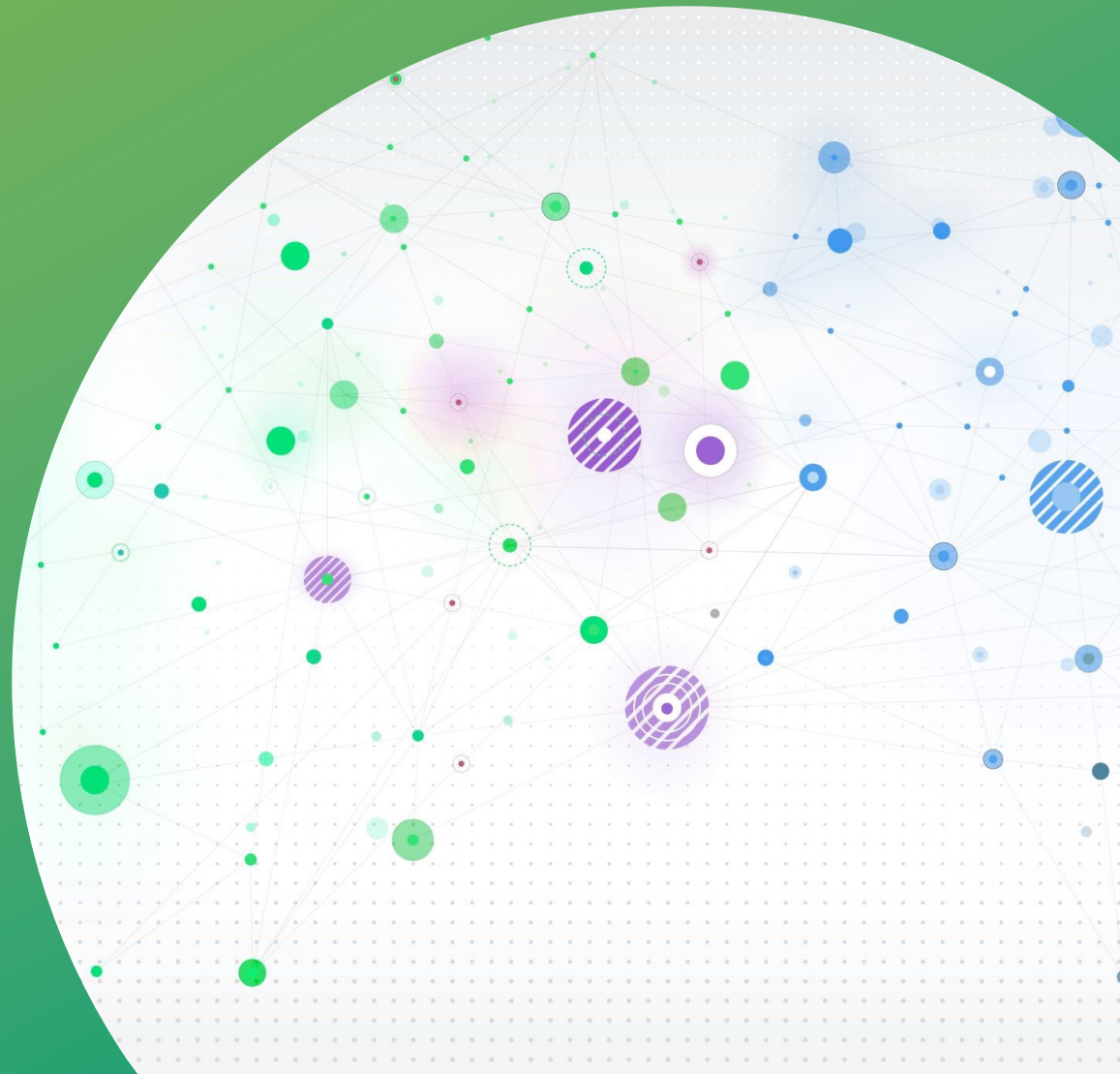
CUSTOMER'S CLUSTERING FOR LARGE MARKET PLACE AND CLUSTERS MAINTENANCE.

Modélisation du comportement client, analyse des clusters dans le temps.



Sommaire.

Introduction.

- Olist est une market-place brésilienne.
- Elle adresse le marché « SMB » des fournisseurs.
- A travers une intégration des flux business, elle propose un support à la logistique, à l'inventaire, à la vente, etc…
- La mission consiste à produire aux équipes d'e-commerce une segmentation des clients qu'elles pourront utiliser au quotidien pour leurs campagnes de communication.
- Il faut en particulier aider les équipes à comprendre les différents types d'utilisateurs. Ces types d'utilisateurs feront l'objet de catégories qui seront utilisées par l'équipe Marketing.
- Il faut également conseiller les équipes marketing quant à la révision des clusters, c'est à dire observer la stabilité des segments au cours du temps.

Phase exploratoire.

- Composition du data-set:
  - Neuf fichiers.

```
df_cust.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 5 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   customer_id               99441 non-null  object
 1   customer_unique_id        99441 non-null  object
 2   customer_zip_code_prefix  99441 non-null  int64 
 3   customer_city             99441 non-null  object
 4   customer_state            99441 non-null  object
dtypes: int64(1), object(4)
memory usage: 3.8+ MB
```

```
df_ord_items.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112650 entries, 0 to 112649
Data columns (total 7 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   order_id             112650 non-null  object 
 1   order_item_id        112650 non-null  int64  
 2   product_id           112650 non-null  object 
 3   seller_id            112650 non-null  object 
 4   shipping_limit_date  112650 non-null  object 
 5   price                112650 non-null  float64
 6   freight_value        112650 non-null  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 6.0+ MB
```

```
df_ord_rev.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99224 entries, 0 to 99223
Data columns (total 7 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   review_id                99224 non-null  object
 1   order_id                 99224 non-null  object
 2   review_score             99224 non-null  int64 
 3   review_comment_title     11568 non-null  object
 4   review_comment_message   40977 non-null  object
 5   review_creation_date     99224 non-null  object
 6   review_answer_timestamp  99224 non-null  object
dtypes: int64(1), object(6)
memory usage: 5.3+ MB
```

```
df_geo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000163 entries, 0 to 1000162
Data columns (total 5 columns):
 #   Column                       Non-Null Count    Dtype  
---  ------                       --------------    -----  
 0   geolocation_zip_code_prefix  1000163 non-null  int64  
 1   geolocation_lat              1000163 non-null  float64
 2   geolocation_lng              1000163 non-null  float64
 3   geolocation_city             1000163 non-null  object 
 4   geolocation_state            1000163 non-null  object 
dtypes: float64(2), int64(1), object(2)
memory usage: 38.2+ MB
```

```
df_ord_pay.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103886 entries, 0 to 103885
Data columns (total 5 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   order_id              103886 non-null  object 
 1   payment_sequential    103886 non-null  int64  
 2   payment_type          103886 non-null  object 
 3   payment_installments  103886 non-null  int64  
 4   payment_value         103886 non-null  float64
dtypes: float64(1), int64(2), object(2)
memory usage: 4.0+ MB
```

```
df_ord_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   order_id                       99441 non-null  object
 1   customer_id                    99441 non-null  object
 2   order_status                   99441 non-null  object
 3   order_purchase_timestamp       99441 non-null  object
 4   order_approved_at              99281 non-null  object
 5   order_delivered_carrier_date   97658 non-null  object
 6   order_delivered_customer_date  96476 non-null  object
 7   order_estimated_delivery_date  99441 non-null  object
dtypes: object(8)
memory usage: 6.1+ MB
```

Phase exploratoire (suite).

```
df_ord_product.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32951 entries, 0 to 32950
Data columns (total 9 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   product_id                  32951 non-null  object 
 1   product_category_name       32341 non-null  object 
 2   product_name_lenght         32341 non-null  float64
 3   product_description_lenght  32341 non-null  float64
 4   product_photos_qty          32341 non-null  float64
 5   product_weight_g            32949 non-null  float64
 6   product_length_cm           32949 non-null  float64
 7   product_height_cm           32949 non-null  float64
 8   product_width_cm            32949 non-null  float64
dtypes: float64(7), object(2)
memory usage: 2.3+ MB
```

```
df_sellers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3095 entries, 0 to 3094
Data columns (total 4 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   seller_id               3095 non-null   object
 1   seller_zip_code_prefix  3095 non-null   int64 
 2   seller_city             3095 non-null   object
 3   seller_state            3095 non-null   object
dtypes: int64(1), object(3)
memory usage: 96.8+ KB
```

```
df_prod_cat.info()
```

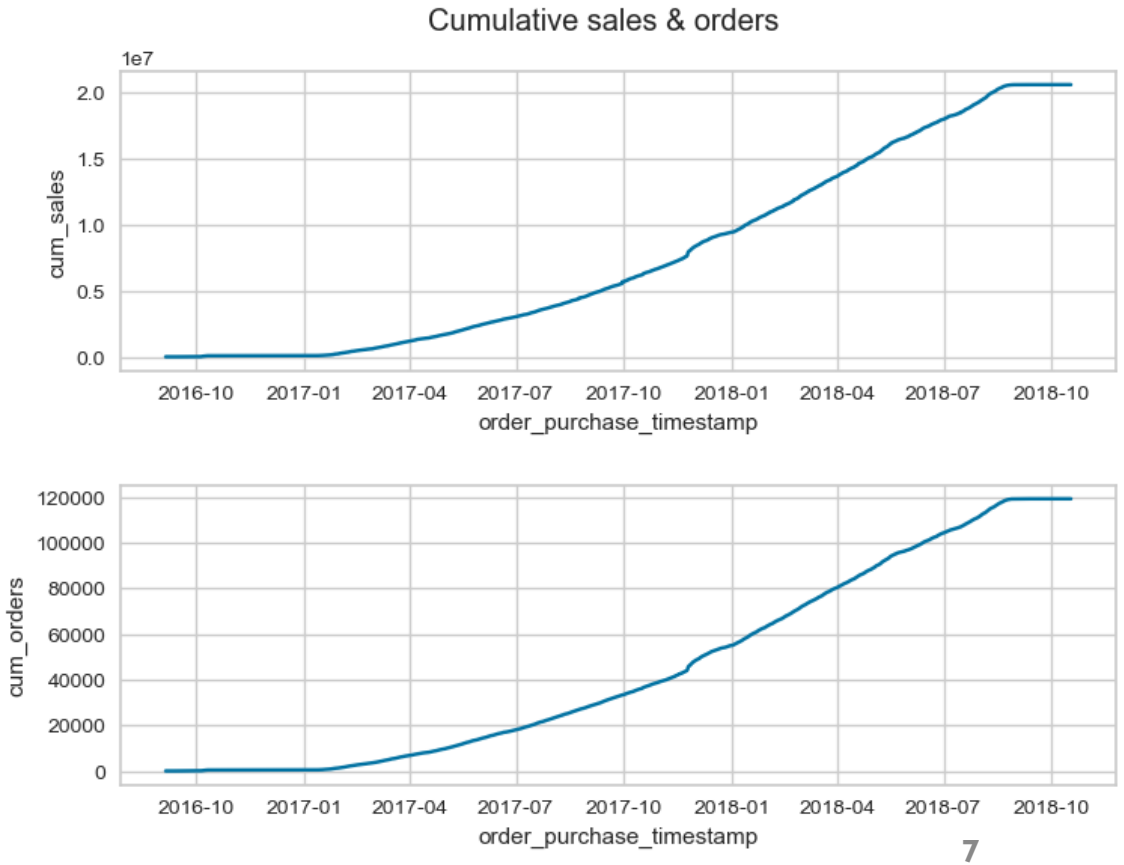
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71 entries, 0 to 70
Data columns (total 2 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   product_category_name          71 non-null     object
 1   product_category_name_english  71 non-null     object
dtypes: object(2)
memory usage: 1.2+ KB
```

Phase exploratoire (suite).

- Opérations successives:
  - « merge » des fichiers.
  - Fichier résultant:
    - 119143 entrées.
    - 40 colonnes.
      - Float64: 15
      - Int64: 1
      - Object: 24
    - Valeurs manquantes: oui.
  - Conversion format.
    - Object → datetime64[ns]
    - Float64, Int64 → string
  - Fichier résultant:
    - 119143 entrées.
    - 40 colonnes.
    - Float64: 13
    - Int64: 0
    - datetime64[ns]: 8
    - Object: 19

Phase exploratoire (suite).

- Aperçu des ventes et commandes.



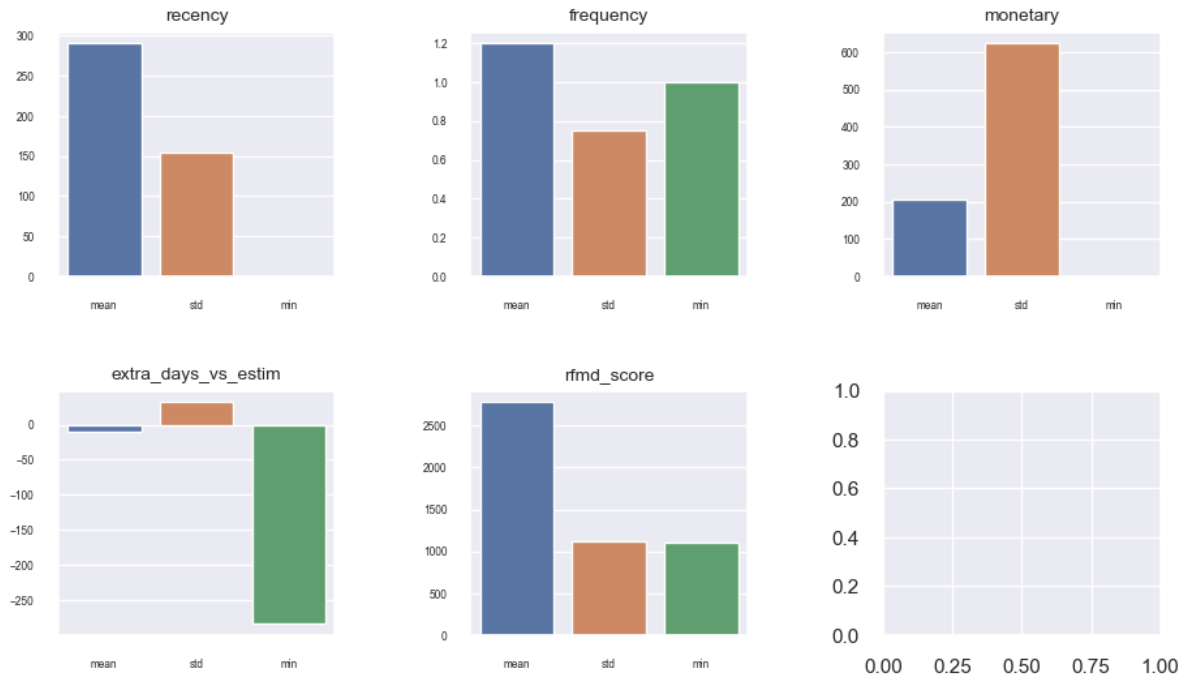
Catégorisation des clients.

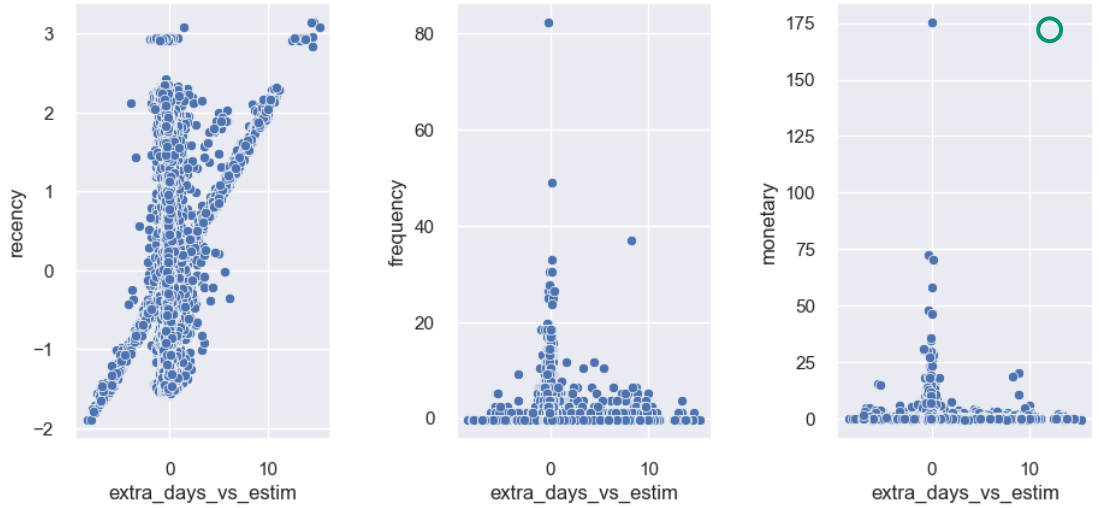
- On propose de catégoriser les clients selon les critères suivants:
  - Récence ou date du dernier achat.
  - Fréquence ou le nombre d'achats réalisés.
  - Montant des achats.
  - Date de livraison versus date de livraison estimée.
- Ces quatre critères doivent permettre de regrouper les clients selon leur score vis à vis de ces critères.
- Plusieurs approches sont possibles pour mesurer les scores.
  - Méthode des quartiles.
  - Méthode des Kmeans.
  - Méthode hiérarchique (dendrogram).
  - Méthode de densité (DBSCAN).
- La visualisation des résultats propres à ces méthodes nécessite une réduction de dimension comme PCA(). Cette méthode est sensible aux valeurs manquantes, ce qui implique des traitements préalables.
- Traitements préalables (preprocessing):
  - Centrage, mise à l'échelle.
  - Remplacement valeurs manquantes par valeurs moyennes.

Catégorisation des clients.

- Méthode des quartiles.
  - Data-set résultant.
  - « rfmd_score_limited » donne un score à chaque client selon ses scores obtenus pour chaque critère.
- Valeurs remarquables/feature.
  - Moyenne.
  - Écart type.
  - Minimum.
  - Écarts-types relativement importants pour les variables recence, fréquence et montant.

customer_id	recency	frequency	monetary	extra_days_vs_estim	r_quartile	f_quartile	m_quartile	d_quartile	rfmd_score	rfmd_score_limited
2a2ce6f8dcda20d059ce98491703	337	1	114.74	-6.0	3	4	2	4	3424	median_low
61a058600d5901f007fab4c27140	458	1	67.41	-10.0	4	4	3	3	4433	bad
1fd6190edaaf884bcaf3d49edf079	596	1	195.42	-16.0	4	4	2	2	4422	bad





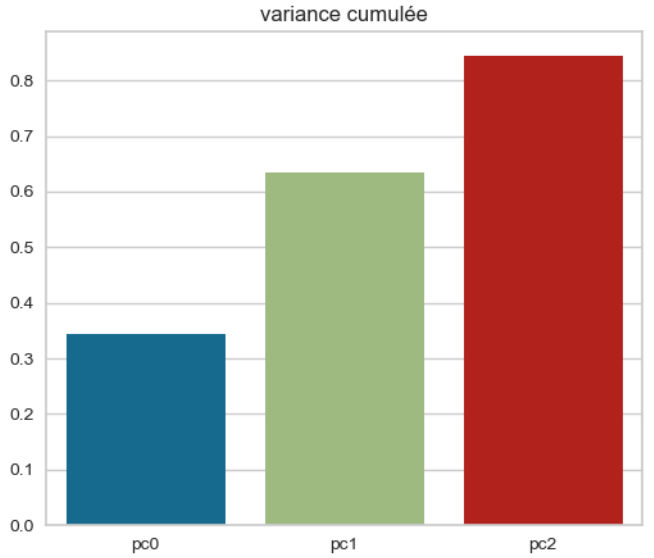
	recency	frequency	monetary	extra_days_vs_estim
0	0.306507	-0.2638	-0.148123	0.115911
1	1.093926	-0.2638	-0.224149	-0.006750
2	1.991974	-0.2638	-0.018526	-0.190742

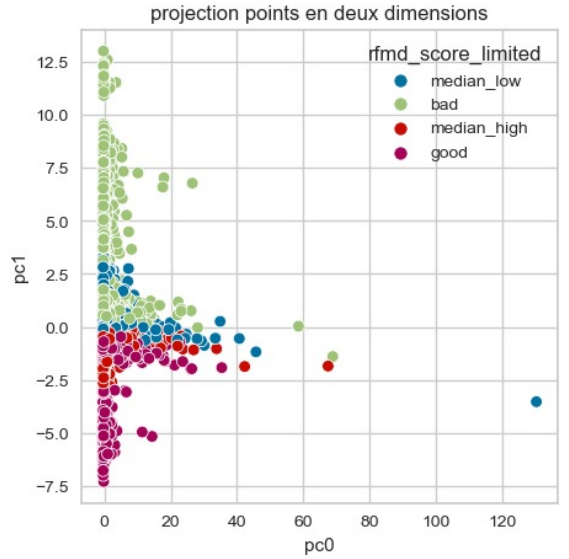
Influence variable délai livraison.

- Est ce que la variable ‘extra_days_vs_estim’ exerce une influence sur les autres variables ?
- On peut visualiser cette influence à partir du data-set RFM.
- Observation:
- Lorsque l'indicateur 'extra_days_vs_estim' est > 0 (c'est à dire un délai de livraison non conforme avec le délai prévu), on observe une dégradation de la récence (de façon linéaire pour une partie des échantillons).
- Lorsque les délais de livraison sont respectés, on observe que la probabilité de fréquence d'achat et de CA augmente.

Visualisation data-set RFM.

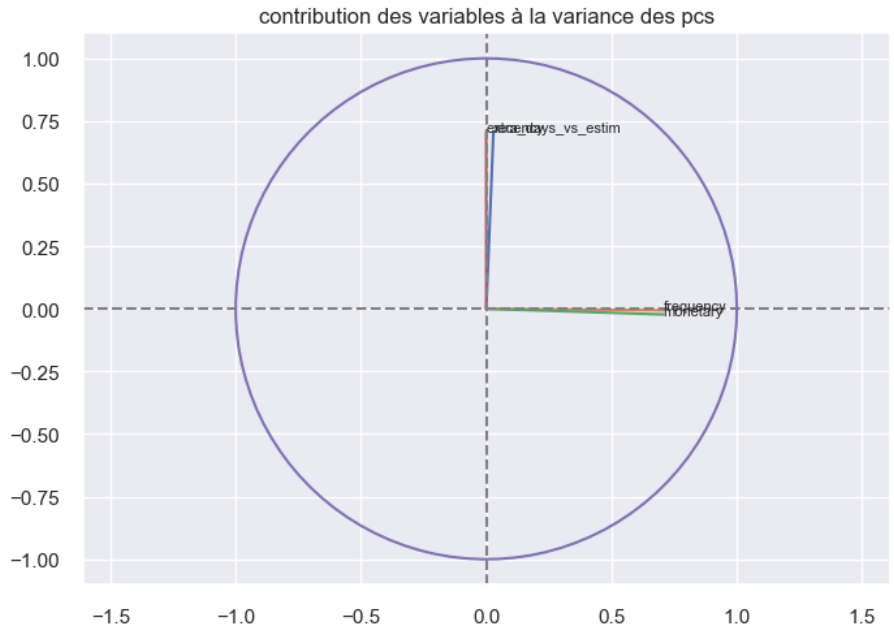
- Après centrage, mise à l'échelle et remplacement valeurs manquantes, méthode de réduction de dimension (PCA) pour visualisation dans un plan à deux dimensions.
- PCA implique une perte d'information.
  - Variance cumulée pour la deuxième composante: 0.64.
- Projection du data-set RFM dans le plan end eux dimensions.





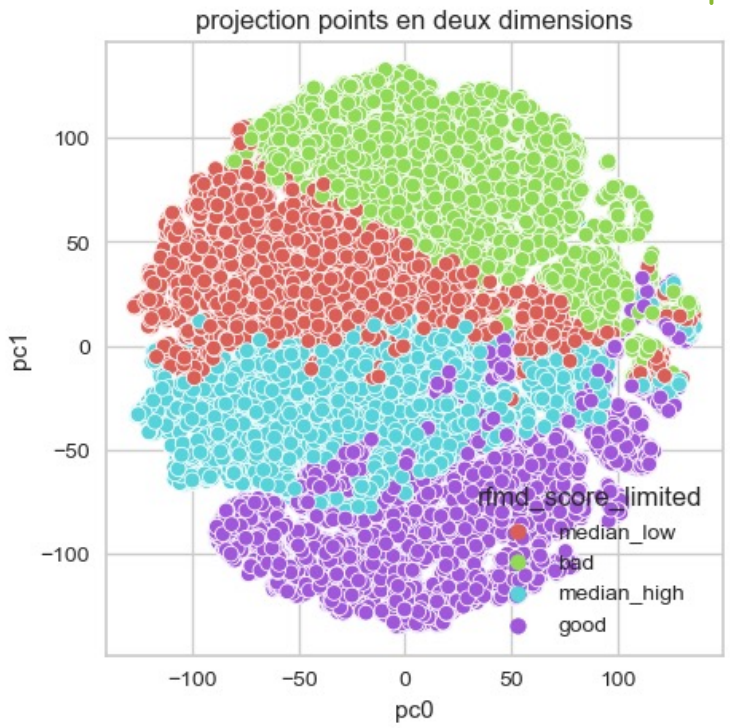
Visualisation data-set RFM (suite).

- Interprétation de la projection en deux dimensions.
  - pc0 représente surtout la variable Montant des achats et fréquence, alors que pc1 représente surtout la variable de la durée de livraison et de la récence.
  - En observant le graphe de la projection des points sur les deux composantes principales, on note que des groupes se dessinent en rapport avec leur projection sur pc1 qui traduit une échelle de valeur des durées de livraison.
    - "Good": projection des points sur pc1 donne des valeurs < 0
    - "Median_high": projection des points sur pc1 donne des valeurs entre 0 et -2.5
    - "Median_low": projection des points sur pc1 donne des valeurs entre 0 et 2.5
    - "Bad": projection des points sur pc1 donne des valeurs supérieures à 2.5.



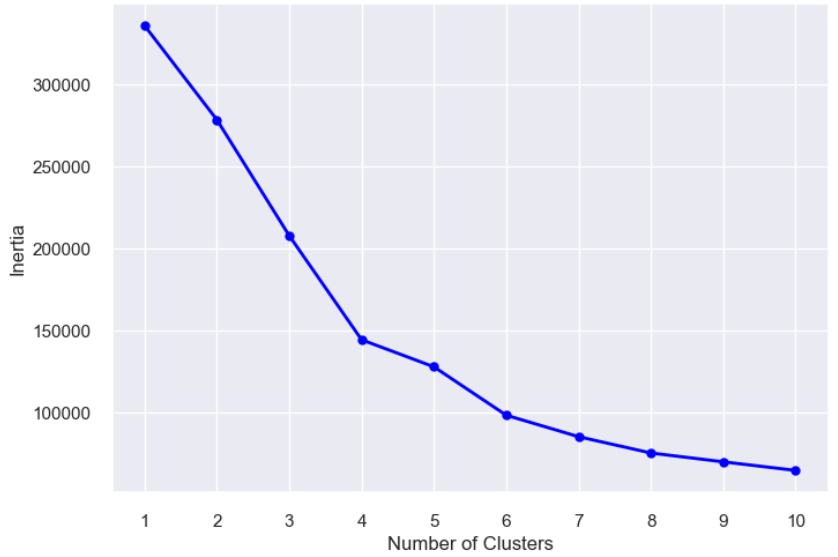
Visualisation data-set RFM (suite).

- Interprétation de la projection avec la méthode de réduction de dimension t-sne.
- Les « bons » clients, en mauve, sont les points situés en deçà de 0 sur l'axe vertical (délais de livraison courts et récence) et plutôt à droite de 0 sur l'axe horizontal (montant, fréquence des achats importants).
- On observe que la fréquence et le montant sont d'autant plus importants que les délais sont respectés.
- On observe l'invserse pour la catégorie « bad ».



Clustering.

- Utilisation de Kmeans() pour élaboration des clusters.
  - Méthode identification nombre de clusters optimal: « elbow method » → 6 clusters.
  - Visualisation clusters.
  - Vérification adéquation nombre de clusters.
    - Silhouette score pour 6 clusters: 0.37.
    - 0.37 montre que la méthode Kmeans a dans une certaine mesure fait un clustering, avec un score plutôt mesuré qui montre que la séparation des clusters est relativement imparfaite.

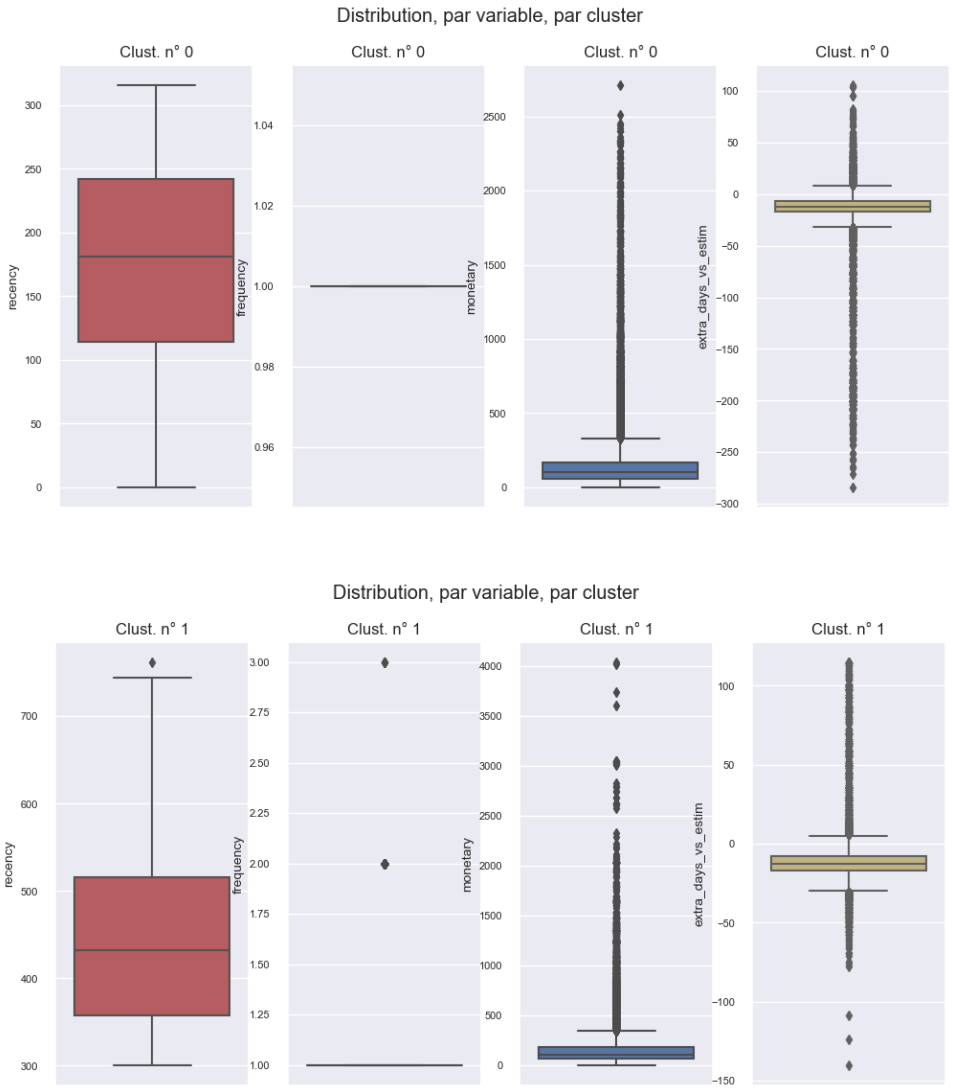


Clustering (suite).

- Interprétation du clustering avec Kmeans.
  - Cluster 3: regroupe clients avec fortes fréquence d'achat et montant.
  - Cluster 0: même profil que 5, mais moins marqué.
  - Cluster 2: regroupe clients avec dates livraison au delà dates estimées.
  - Cluster 1: regroupe clients avec dates livraison en deçà dates estimées.
  - Clusters 4 et 5: regroupent clients avec peu d'achats, montants faibles, dates livraison plus ou moins correctes.

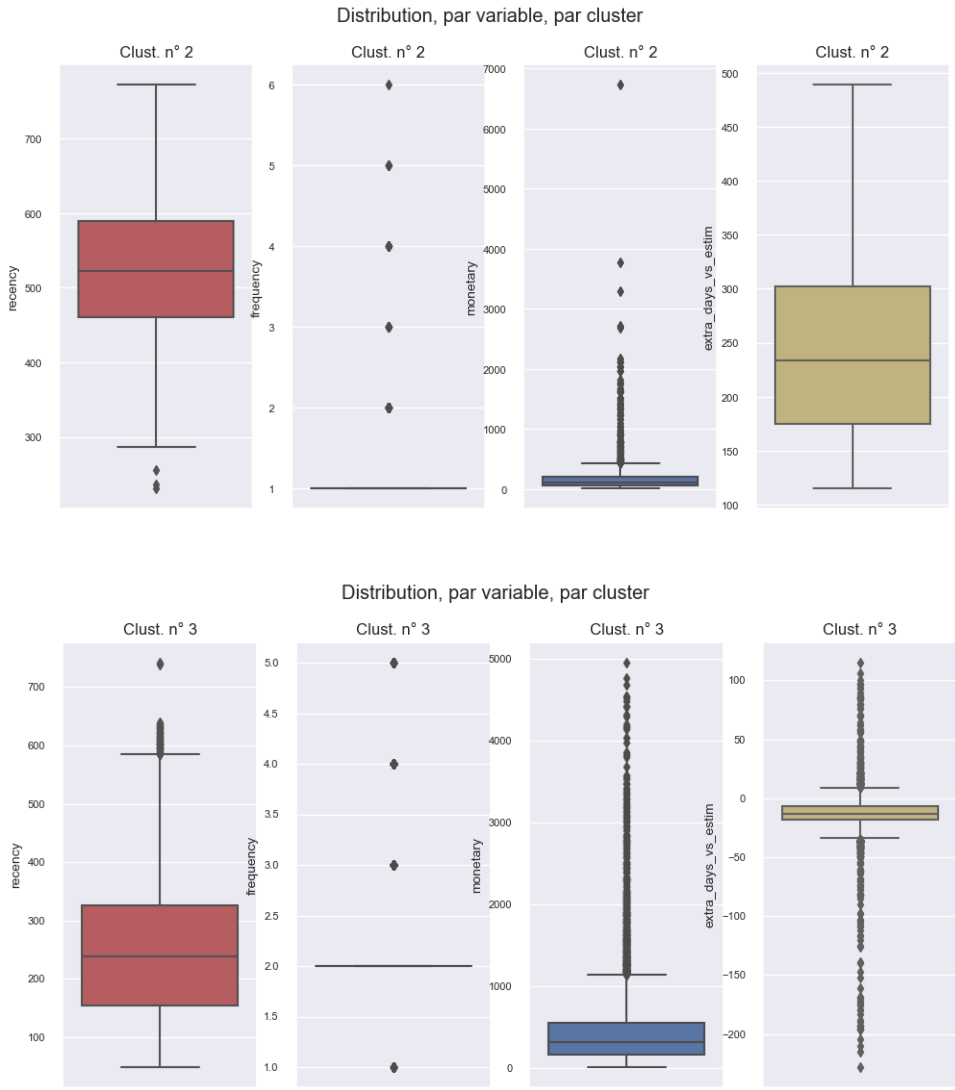
Clustering (suite).

- Distribution de valeurs, par variable, par cluster.
  - Nombre de clients, cluster n° 0: 50175
  - Nombre de clients, cluster n° 1: 37363
  - Nombre de clients, cluster n° 2: 11104
  - Nombre de clients, cluster n° 3: 10004
  - Nombre de clients, cluster n° 4: 17
  - Nombre de clients, cluster n° 5: 778



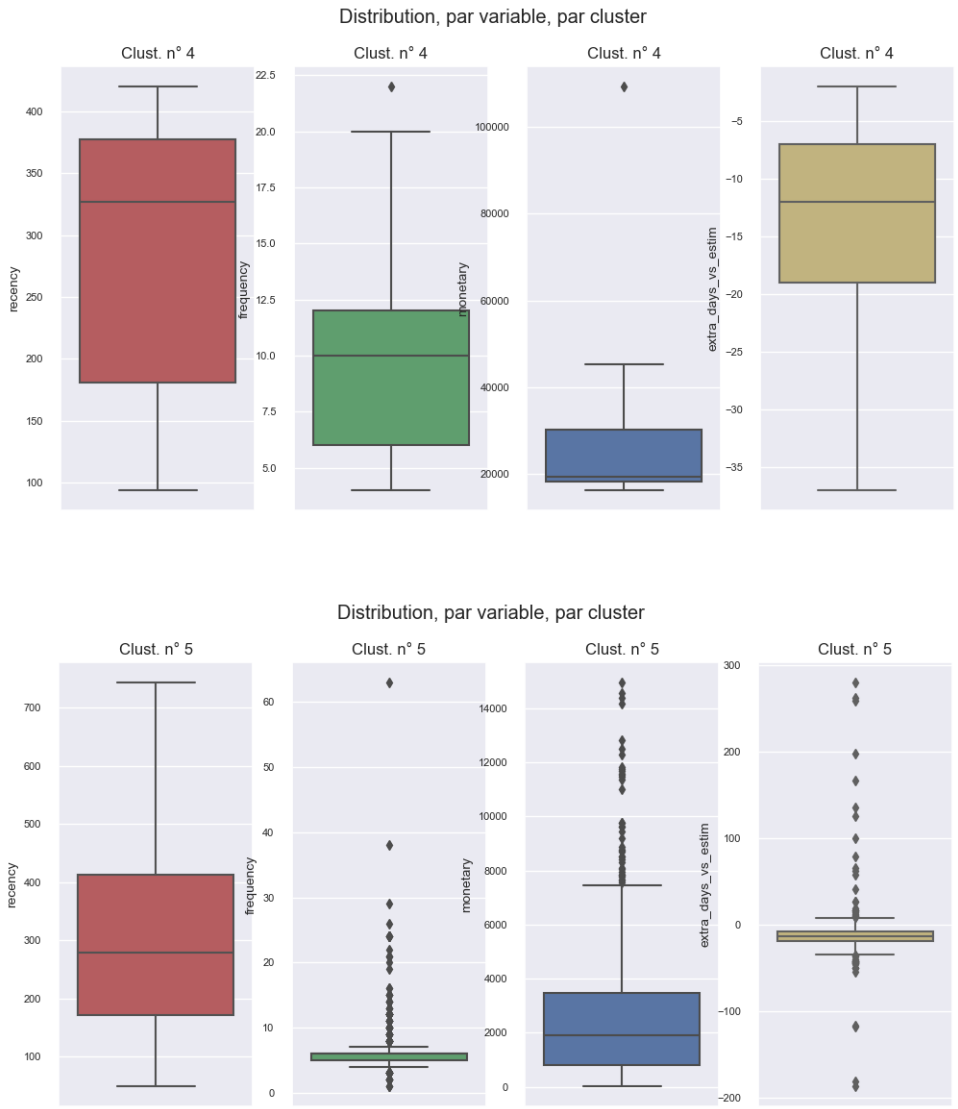
Clustering (suite).

- Clusters 0 et 1.
- Dans les deux cas, bonne qualité de livraison.
- Récence assez mauvaise dans les deux cas (médiane au delà des 180 jours). Encore plus dégradée pour cluster 1.
- Fréquence d'achats faible.
- Montants d'achat à peu près équivalents.



Clustering (suite).

- Clusters 2 et 3.
- Délai mauvais pour cluster 2, bon pour cluster 3.
- Incidence délai sur récence, fréquence, montants.
  - Médiane récence, montants C 2 = 540, 100
  - Médiane récence, montants C 3 = 250, 400

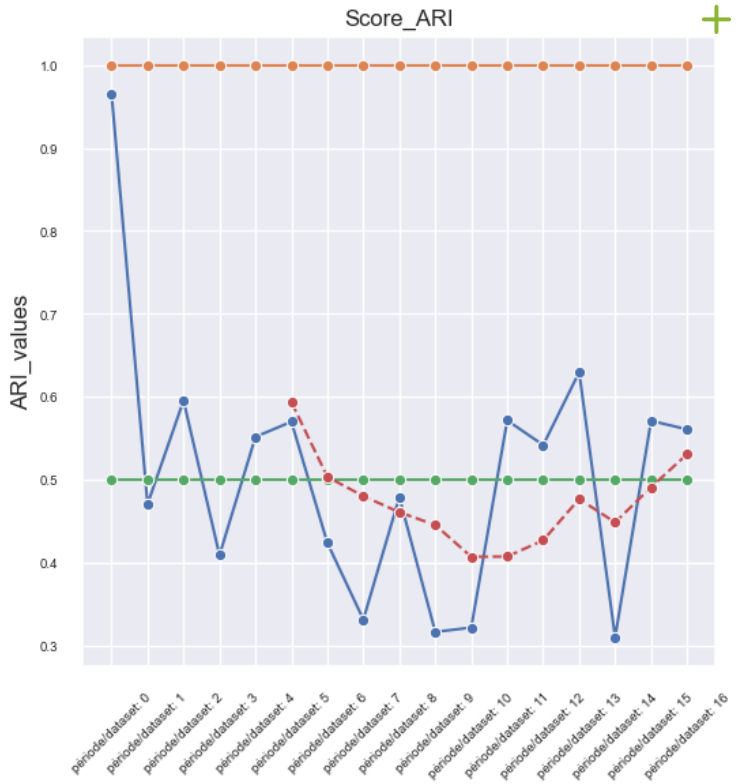


Clustering (suite).

- Clusters 4 et 5.
- Bonne qualité délais.
- Récence importante.
  - Médiane > 280 jours.
- Fréquence.
  - Médianes C 4, C 5: 10, 5.
- Montants.
  - Médianes C 4, C5: 20000, 2000.
- Clusters avec CA les plus importants.

Maintenance du clustering.

- Il est probable que le comportement des clients change en fonction du temps, pour différentes raisons.
- Si c'est le cas à quelle fréquence faudrait-il revoir le clustering ?
- Le graphe ci-contre montre l'évolution du clustering d'un mois sur l'autre.
- Au bout de 3 mois, le score ARI se dégrade fortement (< 0.5).
- La recommandation serait de retravailler le clustering au minimum chaque période de deux mois. Le mieux serait de réduire la période à 1 mois.



CUSTOMER'S CLUSTERING FOR LARGE MARKET PLACE AND CLUSTERS MAINTENANCE.

Fin.