Analyse

July 20, 2022

0.1 Introduction

Nous avons à notre disposition un jeu de données contient des identifiants anonymisés de 4 centres commerciaux français ainsi que des identifiants de téléphones dont les propriétaires ont visité ces centres commerciaux avec la date et l'heure des différents pings observés au cours de la visite. L'objectif de l'étude est, à partir de ces données, de déterminer les horaires d'ouverture de chacun des centres pour chaque jour de la semaine. Idéalement de proposer un algorithme qui permettra de déterminer les horaires d'ouverture d'un centre ne faisant pas partie de ce dataset à partir de ses données de fréquentation; et proposer également une piste d'amélioration pertinente sur la qualité des données source.

```
In [2]: import pandas as pd
        import matplotlib.pyplot as plt
        import numpy as np
        %matplotlib inline
```

0.2 Importation des données

```
In [4]: # Lecture du fichier dans un DataFrame : df
        df = pd.read_csv('/home/sacko/Documents/Data_analyst/Donnees/projet_analytique.csv')
```

0.3 Exploration et nettoyage des données

```
In [61]: df.head(10)
```

Out[61]:

	shopping_center_id	device_local_date	device_hash_id	\
0	b43e9e4f-acd1-4941-874d-e0c5650ab91e	2019-09-14 10:00:25	6fdffac307	
1	b43e9e4f-acd1-4941-874d-e0c5650ab91e	2019-09-14 17:13:15	386141ebd8	
2	b43e9e4f-acd1-4941-874d-e0c5650ab91e	2019-09-14 09:07:06	b06242b848	
3	b43e9e4f-acd1-4941-874d-e0c5650ab91e	2019-09-14 17:14:49	c13cc52e82	
4	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	2019-09-14 10:17:35	f339ddf999	
5	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	2019-09-14 16:52:14	9b491f7bdc	
6	b43e9e4f-acd1-4941-874d-e0c5650ab91e	2019-09-14 17:39:23	d500fae368	
7	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	2019-09-14 12:41:58	62e3eaa686	
8	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	2019-09-14 17:22:23	1255802b0a	
9	599cb959-11ef-49aa-9eb3-e6c17b4ea6ba	2019-09-14 11:34:38	e8396616f9	

day hour

```
0    6    10
1    6    17
2    6     9
3    6    17
4    6    10
5    6    16
6    6    17
7    6    12
8    6    17
9    6    11
```

```
In [53]: df.drop(['Unnamed: 0'], axis=1, inplace = True)
         df.shape
```

```
Out[53]: (81838, 5)
```

```
In [54]: df.isna().sum()
```

```
Out[54]: shopping_center_id    0
         device_local_date     0
         device_hash_id        0
         day                   0
         hour                  0
         dtype: int64
```

```
In [55]: df.describe()
```

```
Out[55]:                 day          hour
         count  81838.000000  81838.000000
         mean       3.566607     14.218761
         std        1.898697      3.701902
         min        1.000000      0.000000
         25%        2.000000     11.000000
         50%        3.000000     14.000000
         75%        5.000000     17.000000
         max        7.000000     23.000000
```

```
In [56]: df.duplicated().sum()
```

```
Out[56]: 3390
```

```
In [58]: df = df.drop_duplicates()
```

```
In [59]: df.shape
```

```
Out[59]: (78448, 5)
```

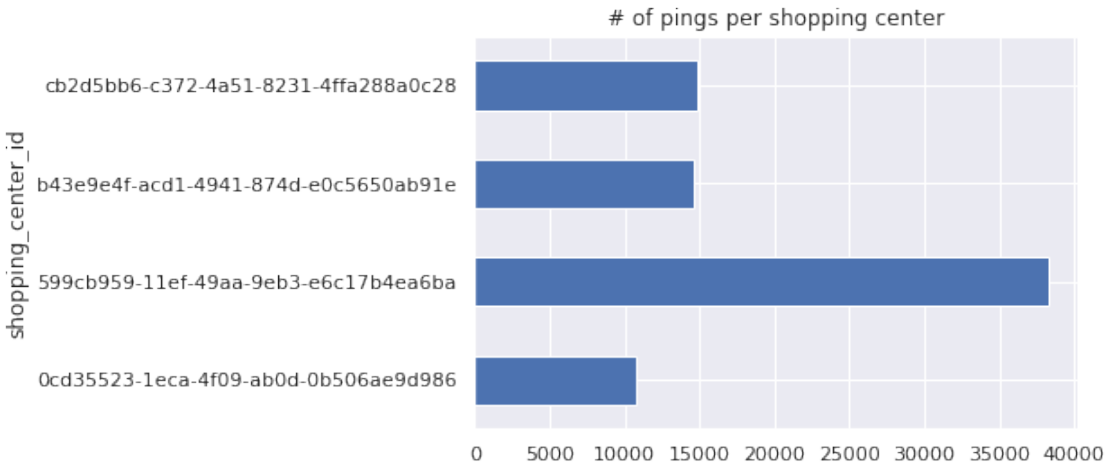
```
In [57]: ## Premiere idee sur les donnees
         print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 81838 entries, 0 to 81837
Data columns (total 5 columns):
shopping_center_id    81838 non-null object
device_local_date     81838 non-null object
device_hash_id        81838 non-null object
day                   81838 non-null int64
hour                  81838 non-null int64
dtypes: int64(2), object(3)
memory usage: 3.1+ MB
None
```

0.4 Évaluation de la quantité de données (de pings) par centre commercial

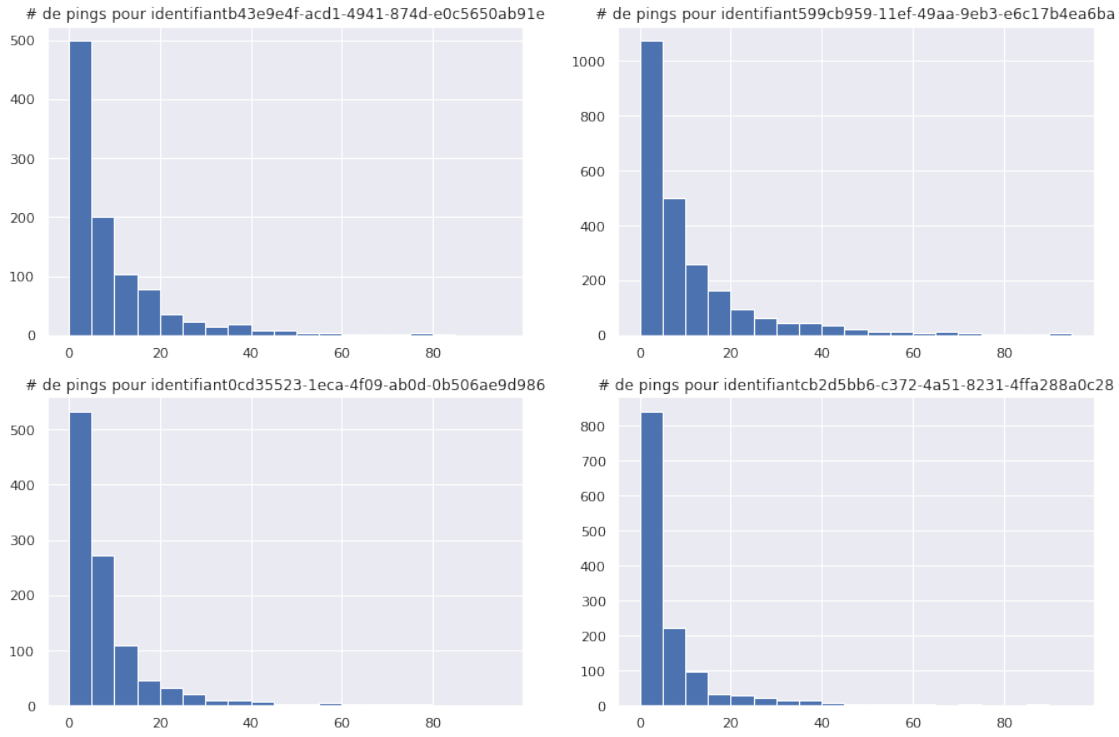
```
In [60]: df.groupby('shopping_center_id')['device_local_date'].count().plot.barh(title='# of pi
```

```
Out[60]: <matplotlib.axes._subplots.AxesSubplot at 0x7f758a853910>
```

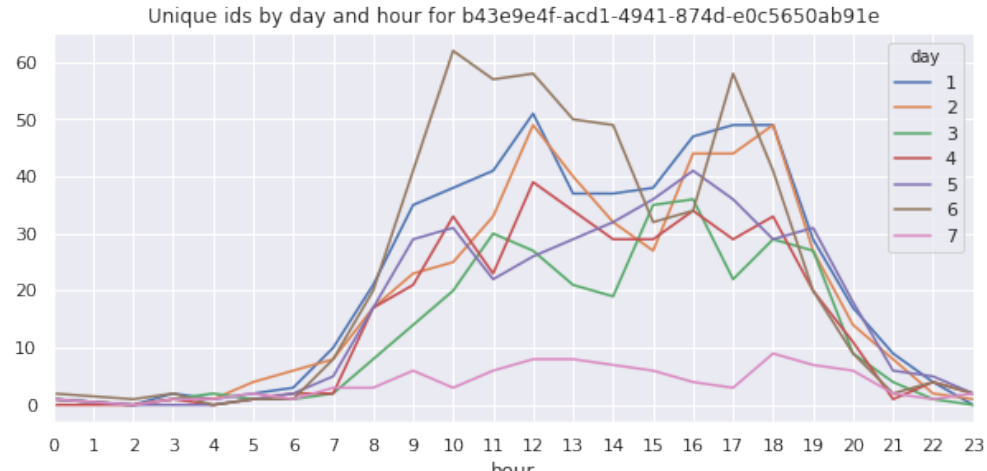


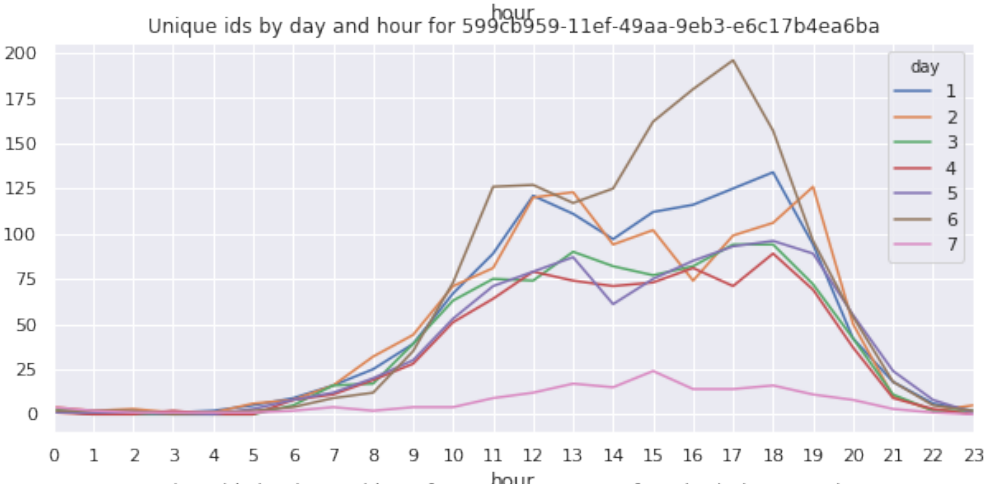
0.5 Évaluation de la qualité des données (de pings par identifiant) par centre commercial

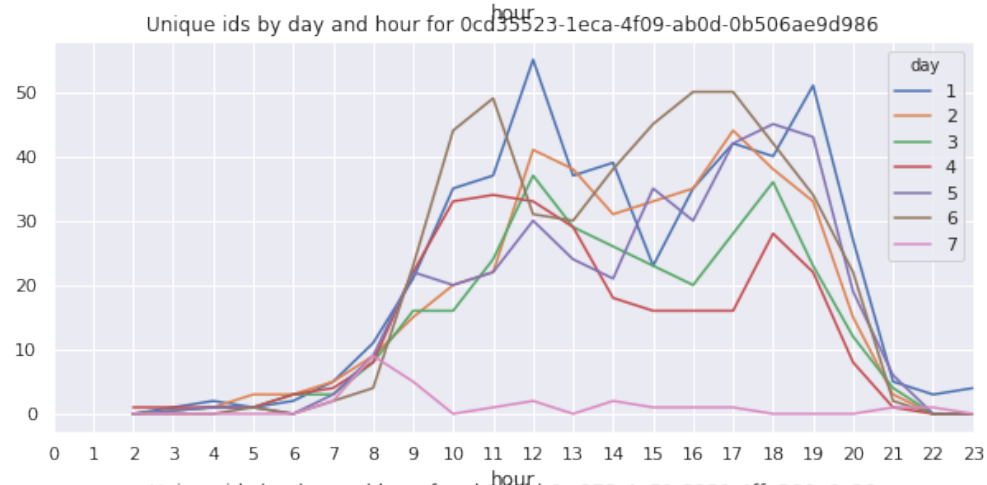
```
In [72]: shopping_centers = df.shopping_center_id.unique()
         pings_per_id = df.groupby(['shopping_center_id', 'device_hash_id'])['device_local_dat

         fig, axs = plt.subplots(2, 2, figsize=(15, 10))

         for shopping_center, ax in zip(shopping_centers, axs.flat):
             pings_per_id.loc[shopping_center].hist(ax=ax, bins=np.arange(0, 100, 5))
             ax.set_title('# de pings pour identifiant' + shopping_center)
```

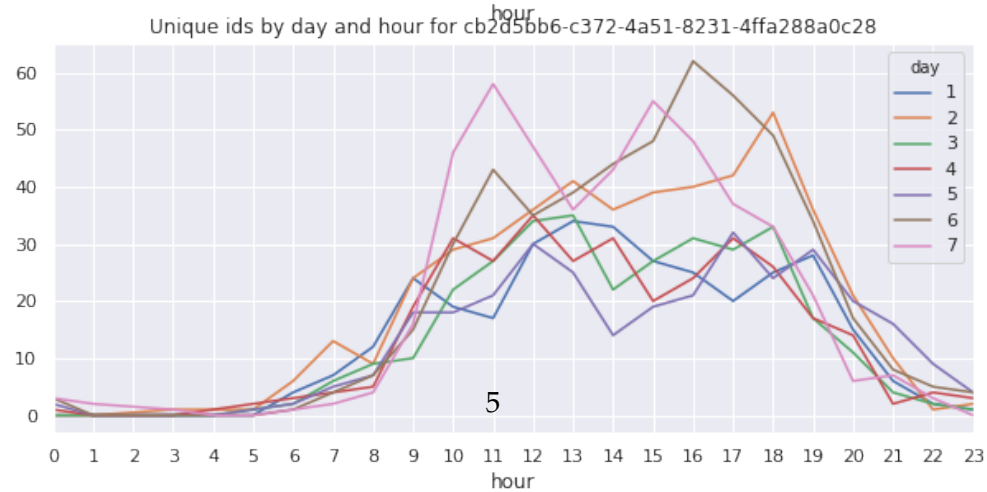



```
In [74]: ids_per_dow_hour = pd.DataFrame(df.groupby(['shopping_center_id', 'day', 'hour'])['dev

         fig, axs = plt.subplots(4, figsize=(10, 20))

         for shopping_center, ax in zip(shopping_centers, axs) :
             ids_per_dow_hour.loc[shopping_center].unstack().T.droplevel(0).sort_index().fillna
             ax.set_title('Unique ids by day and hour for ' + shopping_center)
             ax.set_xticks(np.arange(0, 24, 1))
```


Unique ids by day and hour for b43e9e4f-acd1-4941-874d-e0c5650ab91e
day
1
2
3
4
5
6
7
hour


Unique ids by day and hour for 599cb959-11ef-49aa-9eb3-e6c17b4ea6ba
day
1
2
3
4
5
6
7
hour


Unique ids by day and hour for 0cd35523-1eca-4f09-ab0d-0b506ae9d986
day
1
2
3
4
5
6
7
hour


Unique ids by day and hour for cb2d5bb6-c372-4a51-8231-4ffa288a0c28
day
1
2
3
4
5
6
7
hour

0.6 Conclusion du nettoyage et de l'exploration des données

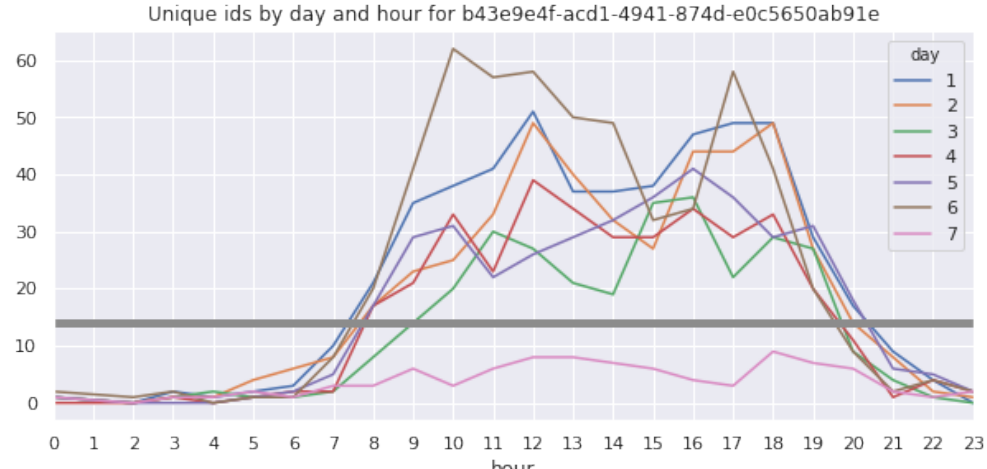
Après avoir supprimé 3390 entrées en double qui correspondent à des personnes ayant émis plusieurs pings dans une minute donnée, le jeu de données contient 78448 pings, représentés par une date et une heure, répartis dans 4 centres commerciaux.

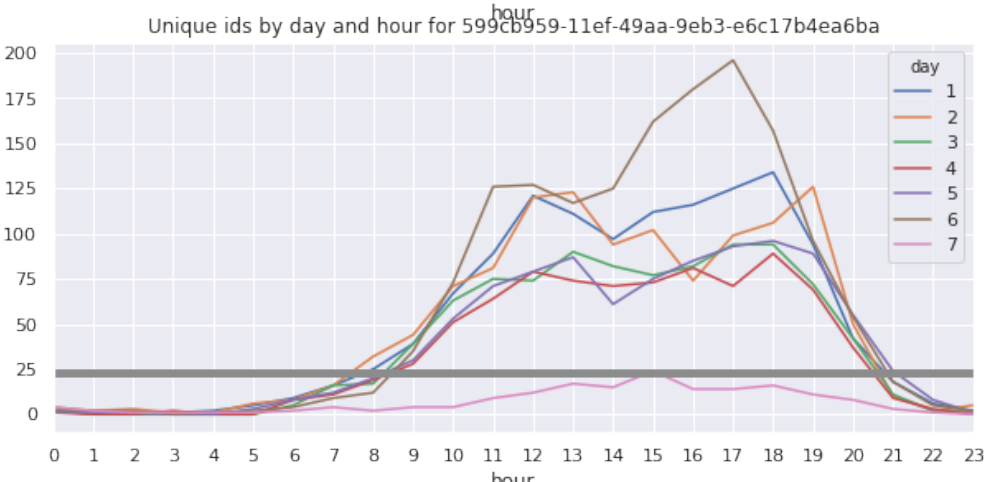
En termes de quantité de données (nombre de pings), l'un des 4 centres commerciaux contient environ 3 fois plus de données que les 3 autres centres commerciaux. En termes de qualité des données (nombre de pings par id), les histogrammes montrent des différences entre les centres qui pourraient avoir un impact sur les analyses avancées, mais qui ne devraient pas poser de problème pour le problème qui nous proccupe, à savoir la détermination des heures d'ouverture des centres commerciaux. La distribution du nombre d'identifiants (qui est un bon indicateur du nombre de personnes présentes dans le centre commercial) jour par jour et heure par heure est cohérente avec notre intuition : les valeurs sont les plus élevées pendant la journée et le samedi, et les plus faibles pendant la nuit et le dimanche, lorsque les centres commerciaux sont généralement fermés (sauf pour 1 des 4 centres commerciaux, qui semble être ouvert le dimanche). En partant du principe que les centres commerciaux sont plus fréquentés lorsqu'ils sont ouverts, nous devrions être en mesure de déterminer les heures d'ouverture des centres commerciaux sur la base de cet ensemble de données.

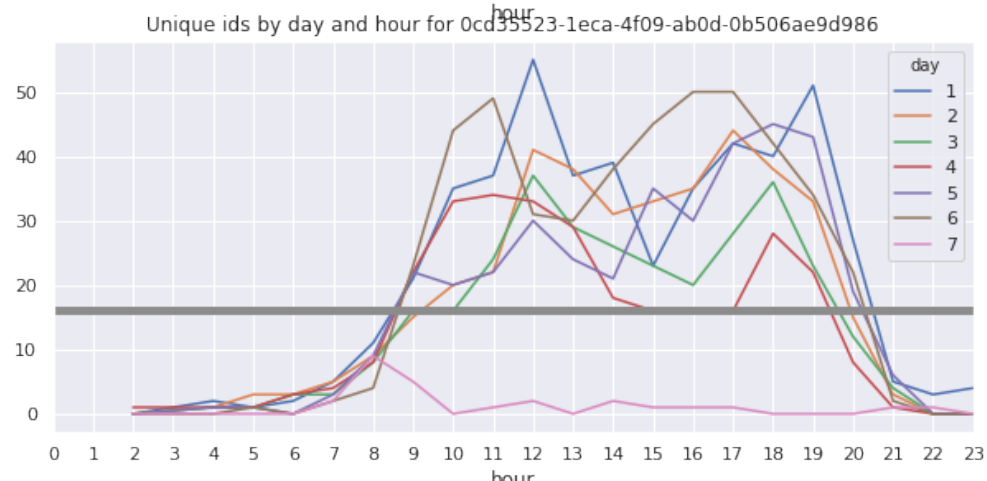
0.7 Détermination des heures d'ouverture des centres commerciaux

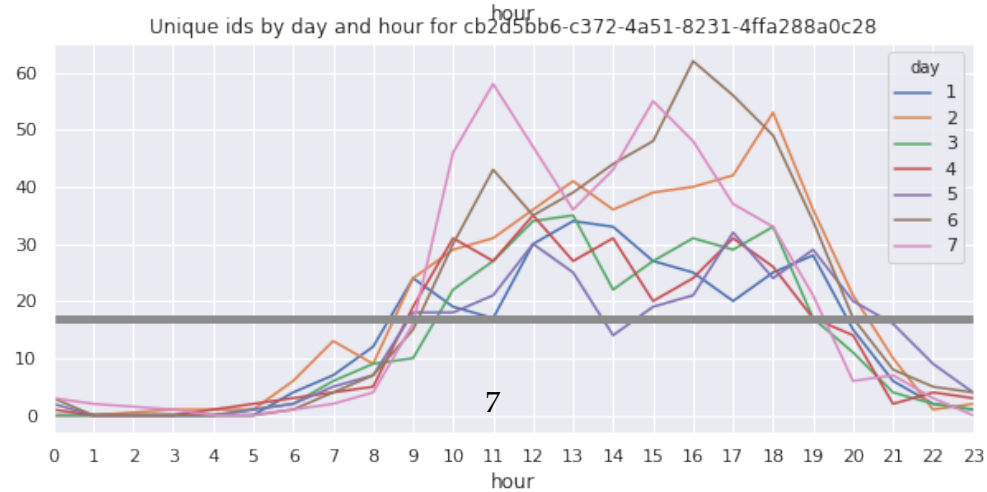
En examinant les graphiques de la phase d'exploration de cette analyse, nous étudierons si le fait de fixer un percentile approprié du nombre d'identifiants dans un centre commercial permet une détection automatique des heures d'ouverture.

```
In [75]: def plot_shopping_center_ids_percentile(percentile=90):
             ids_per_dow_hour = pd.DataFrame(df.groupby(['shopping_center_id', 'day', 'hour'])
             fig, axs = plt.subplots(4, figsize=(10, 20))

             for shopping_center, ax in zip(shopping_centers, axs) :
                 plot_data = ids_per_dow_hour.loc[shopping_center].unstack().T.droplevel(0).so
                 threshold = np.percentile(plot_data, percentile)
                 plot_data.plot(ax=ax)
                 ax.plot([0, 23], [threshold, threshold], linewidth=5)
                 ax.set_title('Unique ids by day and hour for ' + shopping_center)
                 ax.set_xticks(np.arange(0, 24, 1))

In [76]: plot_shopping_center_ids_percentile(55)
```


Unique ids by day and hour for b43e9e4f-acd1-4941-874d-e0c5650ab91e
day
1
2
3
4
5
6
7
hour


Unique ids by day and hour for 599cb959-11ef-49aa-9eb3-e6c17b4ea6ba
day
1
2
3
4
5
6
7
hour


Unique ids by day and hour for 0cd35523-1eca-4f09-ab0d-0b506ae9d986
day
1
2
3
4
5
6
7
hour


Unique ids by day and hour for cb2d5bb6-c372-4a51-8231-4ffa288a0c28
day
1
2
3
4
5
6
7
hour

```
In [77]: def determine_opening_hours(shopping_center_id, percentile=55):
             ids_per_dow_hour = pd.DataFrame(df.groupby(['shopping_center_id', 'day', 'hour'])
             plot_data = ids_per_dow_hour.loc[shopping_center_id].unstack().T.droplevel(0).sor
             threshold = np.percentile(plot_data, percentile)
             res = plot_data > threshold
             return res

In [78]: for c in df.shopping_center_id.unique() :
             print('\n-----------------------------------------------------\n'+ c + '\n')
             print(determine_opening_hours(c))
```

```
-----------------------------------------------------
b43e9e4f-acd1-4941-874d-e0c5650ab91e

day       1      2      3      4      5      6      7
hour
0     False  False  False  False  False  False  False
2     False  False  False  False  False  False  False
3     False  False  False  False  False  False  False
4     False  False  False  False  False  False  False
5     False  False  False  False  False  False  False
6     False  False  False  False  False  False  False
7     False  False  False  False  False  False  False
8      True   True  False   True   True   True  False
9      True   True  False   True   True   True  False
10     True   True   True   True   True   True  False
11     True   True   True   True   True   True  False
12     True   True   True   True   True   True  False
13     True   True   True   True   True   True  False
14     True   True   True   True   True   True  False
15     True   True   True   True   True   True  False
16     True   True   True   True   True   True  False
17     True   True   True   True   True   True  False
18     True   True   True   True   True   True  False
19     True   True   True   True   True   True  False
20     True  False  False  False   True  False  False
21    False  False  False  False  False  False  False
22    False  False  False  False  False  False  False
23    False  False  False  False  False  False  False

-----------------------------------------------------
599cb959-11ef-49aa-9eb3-e6c17b4ea6ba

day       1      2      3      4      5      6      7
```

hour							
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False
8	True	True	False	False	False	False	False
9	True	True	True	True	True	True	False
10	True	True	True	True	True	True	False
11	True	True	True	True	True	True	False
12	True	True	True	True	True	True	False
13	True	True	True	True	True	True	False
14	True	True	True	True	True	True	False
15	True	True	True	True	True	True	True
16	True	True	True	True	True	True	False
17	True	True	True	True	True	True	False
18	True	True	True	True	True	True	False
19	True	True	True	True	True	True	False
20	True	True	True	True	True	True	False
21	False	False	False	False	True	False	False
22	False	False	False	False	False	False	False
23	False	False	False	False	False	False	False

0cd35523-1eca-4f09-ab0d-0b506ae9d986

day	1	2	3	4	5	6	7
hour							
2	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False
9	True	False	False	True	True	True	False
10	True	True	False	True	True	True	False
11	True	True	True	True	True	True	False
12	True	True	True	True	True	True	False
13	True	True	True	True	True	True	False
14	True	True	True	True	True	True	False
15	True	True	True	False	True	True	False
16	True	True	True	False	True	True	False
17	True	True	True	False	True	True	False
18	True	True	True	True	True	True	False
19	True	True	True	True	True	True	False

```
20     True  False  False  False   True   True  False
21    False  False  False  False  False  False  False
22    False  False  False  False  False  False  False
23    False  False  False  False  False  False  False

--------------------------------------------------
cb2d5bb6-c372-4a51-8231-4ffa288a0c28

day       1      2      3      4      5      6      7
hour
0     False  False  False  False  False  False  False
1     False  False  False  False  False  False  False
3     False  False  False  False  False  False  False
4     False  False  False  False  False  False  False
5     False  False  False  False  False  False  False
6     False  False  False  False  False  False  False
7     False  False  False  False  False  False  False
8     False  False  False  False  False  False  False
9      True   True  False   True   True  False  False
10     True   True   True   True   True   True   True
11    False   True   True   True   True   True   True
12     True   True   True   True   True   True   True
13     True   True   True   True   True   True   True
14     True   True   True   True  False   True   True
15     True   True   True   True   True   True   True
16     True   True   True   True   True   True   True
17     True   True   True   True   True   True   True
18     True   True   True   True   True   True   True
19     True   True  False  False   True   True   True
20    False   True  False  False   True  False  False
21    False  False  False  False  False  False  False
22    False  False  False  False  False  False  False
23    False  False  False  False  False  False  False
```

0.8 Conclusion

Nous avons développé un algorithme qui permet de détecter les heures d'ouverture des centres commerciaux à partir des pings des téléphones portables. Notre algorithme est basé sur un seuil, défini comme le 55 ème centile du nombre d'identifiants détectés en une heure sur une semaine standard dans un centre commercial donné. Les faiblesses suivantes et les améliorations possibles sont soulignées: Il existe des pings d'employés qui sont éventuellement présents en dehors des heures d'ouverture. Ils pourraient être filtrés en détectant les identifiants présents pendant de très longues périodes de temps un jour donné, ou les personnes présentes plusieurs jours dans la semaine. Notre modèle est basé sur seulement 4 centres commerciaux. Afin de construire un modèle plus robuste, nous devrions le tester sur un plus grand nombre de centres commerciaux. Nous pourrions vérifier les heures d'ouverture réelles (sur internet, en appelant le centre commercial) et réajuster notre modèle afin de coller au mieux à la réalité.