

Année scolaire 2016-2017

Projet de technologies NoSQL - Le meilleur endroit où s'installer à New York

ÉLÈVE

M^r Olivier Yoo

Sommaire

1	Définition du "meilleur endroit"	2
2	Les données	3
2.1	Description des données utilisées	3
2.2	En pratique	4
3	À la recherche de l'endroit idéal	5
3.1	La méthodologie utilisée	5
3.2	En pratique	5
3.2.1	Import des bases sur MongoDB	5
3.2.2	Calcul des classements	6
3.2.3	Agrégation des classements	6
3.2.4	Et le gagnant est...	7
4	Discussion et axes d'améliorations	9
4.1	Le choix du type de base de données	9
4.2	L'influence de la taille	9
4.3	L'utilisation des coordonnées spatiales.	9

Chapitre 1

Définition du "meilleur endroit"

Plus qu'une coordonnée (x, y) précise, lorsqu'une personne cherche à s'installer quelque part elle cherche une zone, un quartier, un arrondissement, dans lequel il se sentirait à l'aise. C'est pourquoi dans ce projet nous essaierons plus de déterminer une "zone idéale" plutôt que des "coordonnées idéale". En effet, quand on demande à un agent immobilier de nous trouver l'appartement de nos rêves, il est plus commun de lui faire part des quartiers qui nous intéressent plutôt que de sa latitude et longitude favorite. Dans le cas de la ville de New York nous allons donc tout d'abord essayer de déterminer l'arrondissement qui parait le plus intéressant selon de nombreux critères (environnement, éducation, culture, santé, etc..). Une fois l'arrondissement choisi, on sélectionnera quelques quartiers pouvant nous intéresser.

Chapitre 2

Les données

2.1 Description des données utilisées

Les données proviennent du [portail open data](#) fournit par la ville de New-York. Pour mener à bien ce projet j'ai décidé d'importer 8 bases de données pouvant se décomposer en plusieurs catégories.

Education

- [La taille de la classe](#) : Cette base de données référence toute les écoles de New-York ainsi que la taille moyenne de leurs classes. Comme nous ne voulons pas que nos enfants travaillent dans des classes trop surchargées nous avons décidé de prendre ce paramètre en considération.
- [Les résultats scolaires](#) : Cette base sera utilisée afin de savoir quel arrondissement à en moyenne les meilleurs résultats scolaires.

Santé

[Les hopitaux](#) : Cette base sera utilisée afin de déterminer l'arrondissement avec le plus d'hopitaux.

Culture

[Les principaux monuments](#) : Cette base sera utilisée afin de déterminer l'arrondissement avec le plus de monuments historiques.

Environnement

[Les plaintes sur la qualité de l'eau](#) : Cette base sera utilisée afin de déterminer l'arrondissement avec le moins de plaintes sur la qualité de l'eau.

Social

[Les hotspots Wi-Fi](#) Cette base sera utilisée afin de déterminer l'arrondissement ayant le plus de hotspots Wi-Fi.

Sécurité

[Les collisions](#) Cette base sera utilisée afin de déterminer l'arrondissement ayant le moins de collisions.

Ces sept bases nous permettront de déterminer l'arrondissement le plus intéressant.

Finalement, la base sur [le nombre d'habitants par quartier](#) nous permettra de trouver ce petit coin de tranquillité au beau milieu du "meilleur arrondissement de New-York".

2.2 En pratique

Placez-vous à l'endroit où vous souhaitez créer le projet et lancez la ligne de code suivante dans un terminal :

```
bash Recup_Donnees
```

Ce script va créer un dossier nommé *Projet_Olivier_Yoo* et va télécharger les données à l'intérieur de ce dossier.

```

olivier@olivier-HP-ZBook-15: ~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL
olivier@olivier-HP-ZBook-15:~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL$ bash Recup_Donnees
rm: Impossible de supprimer 'Projet_Olivier_Yoo/': Aucun fichier ou dossier de ce type
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 13218 0 13218 0 12804 0 --:--:-- 0:00:01 --:--:-- 21216
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 19826 0 19826 0 25172 0 --:--:-- 0:00:01 --:--:-- 25641
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 59113 0 59113 0 50747 0 --:--:-- 0:00:01 --:--:-- 56881
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 36487 0 36487 0 47802 0 --:--:-- 0:00:01 --:--:-- 48844
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 4108k 0 4108k 0 573k 0 --:--:-- 0:00:07 --:--:-- 688k
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 177M 0 177M 0 1278k 0 --:--:-- 0:02:22 --:--:-- 853k
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 14.2M 0 14.2M 0 1150k 0 --:--:-- 0:00:12 --:--:-- 1391k
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 516k 0 516k 0 368k 0 --:--:-- 0:00:01 --:--:-- 372k
olivier@olivier-HP-ZBook-15:~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL$

```

Chapitre 3

À la recherche de l'endroit idéal

3.1 La méthodologie utilisée

La méthodologie utilisée se décompose comme ci-dessous :

1. Pour chacune des bases utilisées, on effectue un classement des arrondissements en fonction du critère d'intérêt.
2. On agrège les classements de manière appropriée afin d'obtenir "l'arrondissement idéal".
3. On détermine les quartiers les plus intéressants au sein de cet "arrondissement idéal".

3.2 En pratique

3.2.1 Import des bases sur MongoDB

Tout d'abord, assurez-vous d'avoir télécharger le client mongo à l'aide de la commande :

sudo apt install mongodb-clients Puis, dans le même terminal que dans le chapitre précédent, lancez la ligne de code suivante :

bash Import_Mongo

Ce script se chargera d'importer les bases dans MongoDB.

Remarque : Il faut s'assurer que le serveur Mongo est disponible sur le port 3333 du localhost.

```
olivier@olivier-HP-ZBook-15: ~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL
olivier@olivier-HP-ZBook-15:~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL$ bash Import_Mongo
[sudo] Mot de passe de olivier :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
mongodb-clients is already the newest version (1:2.6.10-0ubuntu1).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
linux-headers-4.4.0-42 linux-headers-4.4.0-42-generic linux-headers-4.4.0-45 linux-headers-4.4.0-47 linux-headers-4.4.0-47-generic linux-headers-4.4.0-53
linux-headers-4.4.0-53-generic linux-headers-4.4.0-57 linux-headers-4.4.0-57-generic linux-image-4.4.0-42-generic linux-image-4.4.0-45-generic linux-image-4.4.0-47-generic
linux-image-4.4.0-53-generic linux-image-extra-4.4.0-42-generic linux-image-extra-4.4.0-45-generic linux-image-extra-4.4.0-47-generic linux-image-extra-4.4.0-53-generic
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 170 non mis à jour.
connected to 127.0.0.1:3333
no collection specified!
using filename 'Hospitals_Corporation_Facilities' as collection.
2017-02-16T19:38:23.442+0100 dropping: projetNoSQL.Hospitals_Corporation_Facilities
2017-02-16T19:38:23.481+0100 check 9 79
2017-02-16T19:38:23.482+0100 imported 78 objects
connected to 127.0.0.1:3333
no collection specified!
using filename 'Population_By_Neighborhood' as collection.
2017-02-16T19:38:23.506+0100 dropping: projetNoSQL.Population_By_Neighborhood
2017-02-16T19:38:23.568+0100 check 9 391
2017-02-16T19:38:23.573+0100 imported 390 objects
connected to 127.0.0.1:3333
no collection specified!
using filename 'Graduation_Outcomes' as collection.
2017-02-16T19:38:23.594+0100 dropping: projetNoSQL.Graduation_Outcomes
2017-02-16T19:38:23.660+0100 check 9 386
2017-02-16T19:38:23.666+0100 imported 385 objects
connected to 127.0.0.1:3333
no collection specified!
using filename 'Class_Size' as collection.
2017-02-16T19:38:23.689+0100 dropping: projetNoSQL.Class_Size
2017-02-16T19:38:23.775+0100 check 9 659
2017-02-16T19:38:23.780+0100 imported 658 objects
connected to 127.0.0.1:3333
no collection specified!
using filename 'Water_Complaint_Street' as collection.
2017-02-16T19:38:23.802+0100 dropping: projetNoSQL.Water_Complaint_Street
2017-02-16T19:38:23.823+0100 Progress: 1447269/4195944 34%
2017-02-16T19:38:23.823+0100 Progress: 2800 82/second
2017-02-16T19:39:19.172+0100 Progress: 1919999/4195944 45%
2017-02-16T19:39:19.172+0100 Progress: 3700 60/second
2017-02-16T19:39:53.736+0100 Progress: 2390847/4195944 56%
2017-02-16T19:39:53.736+0100 Progress: 4600 51/second
2017-02-16T19:40:24.251+0100 Progress: 2806111/4195944 66%
2017-02-16T19:40:24.251+0100 Progress: 5400 44/second
2017-02-16T19:40:49.345+0100 Progress: 3272541/4195944 77%
2017-02-16T19:40:49.345+0100 Progress: 6300 43/second
2017-02-16T19:41:19.907+0100 Progress: 3733428/4195944 88%
2017-02-16T19:41:19.907+0100 Progress: 7200 40/second
2017-02-16T19:41:53.975+0100 check 9 8095
2017-02-16T19:42:40.066+0100 imported 8094 objects
connected to 127.0.0.1:3333
no collection specified!
using filename 'Landmarks' as collection.
2017-02-16T19:42:40.086+0100 dropping: projetNoSQL.Landmarks
2017-02-16T19:42:41.140+0100 check 9 47505
2017-02-16T19:42:41.285+0100 imported 47504 objects
connected to 127.0.0.1:3333
```

3.2.2 Calcul des classements

Le classement pour chacune des collections est disponible en exécutant chacune de ces lignes de code dans un terminal (**Attention** lors du copier coller de la ligne de code, remplacez le "-" devant "port" par un double tiret, cf image ci-dessous) :

- mongo --port 3333 projetNoSQL < Class_Size.js
- mongo --port 3333 projetNoSQL < Graduation_Outcomes.js
- mongo --port 3333 projetNoSQL < Hospitals_Corporation_Facilities.js
- mongo --port 3333 projetNoSQL < Landmarks.js
- mongo --port 3333 projetNoSQL < Motor_Vehicle_Collisions.js
- mongo --port 3333 projetNoSQL < Water_Complaint_Street.js
- mongo --port 3333 projetNoSQL < Wifi.js

```

olivier@olivier-HP-ZBook-15: ~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL
MongoDB shell version: 2.6.10
connecting to: 127.0.0.1:3333/projetNoSQL
Classement du nombre moyen d'élèves par classe
{ "_id": "Manhattan", "mean": 21.184476119403 }
{ "_id": "Bronx", "mean": 21.36350364963504 }
{ "_id": "Brooklyn", "mean": 21.464788732394368 }
{ "_id": "Queens", "mean": 22.74701492573133 }
{ "_id": "Staten Island", "mean": 23.7 }
bye

olivier@olivier-HP-ZBook-15: ~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL mongo --port 3333 projetNoSQL < Graduation_Outcomes.js
MongoDB shell version: 2.6.10
connecting to: 127.0.0.1:3333/projetNoSQL
Classement du pourcentage moyen d'élèves diplômés par école
{ "_id": "Staten Island", "ratio": 0.62718294808047 }
{ "_id": "Manhattan", "ratio": 0.580821711567866 }
{ "_id": "Queens", "ratio": 0.5564621065445703 }
{ "_id": "Brooklyn", "ratio": 0.5191808364908987 }
{ "_id": "Bronx", "ratio": 0.50872791603955 }
Classement du pourcentage moyen d'élèves recalés par école
{ "_id": "Staten Island", "ratio": 0.11050665573643448 }
{ "_id": "Manhattan", "ratio": 0.1279522776293588 }
{ "_id": "Queens", "ratio": 0.1491520187916984 }
{ "_id": "Brooklyn", "ratio": 0.1550164264836901 }
{ "_id": "Bronx", "ratio": 0.16717436247644854 }
bye

olivier@olivier-HP-ZBook-15: ~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL mongo --port 3333 projetNoSQL < Hospitals_Corporation_Facilities.js
MongoDB shell version: 2.6.10
connecting to: 127.0.0.1:3333/projetNoSQL
Classement des arrondissements en fonction du nombre d'hopitaux
{ "_id": "Brooklyn", "cun": 26 }
{ "_id": "Manhattan", "cun": 24 }
{ "_id": "Bronx", "cun": 14 }
{ "_id": "Queens", "cun": 11 }
{ "_id": "Staten Island", "cun": 3 }
bye

olivier@olivier-HP-ZBook-15: ~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL mongo --port 3333 projetNoSQL < Landmarks.js
MongoDB shell version: 2.6.10
connecting to: 127.0.0.1:3333/projetNoSQL
Classement des arrondissements en fonction du nombre de monuments historiques
{ "_id": "MN", "cun": 21190 }
{ "_id": "BK", "cun": 18860 }
{ "_id": "QM", "cun": 5860 }
{ "_id": "BX", "cun": 1263 }
{ "_id": "SI", "cun": 1120 }
{ "_id": "", "cun": 11 }
bye

olivier@olivier-HP-ZBook-15: ~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL mongo --port 3333 projetNoSQL < Motor_Vehicle_Collisions.js
MongoDB shell version: 2.6.10
connecting to: 127.0.0.1:3333/projetNoSQL
Classement des arrondissements en fonction du nombre d'accidents de la route
{ "_id": "STATEN ISLAND", "cun": 33586 }
{ "_id": "BRONX", "cun": 94130 }
{ "_id": "MANHATTAN", "cun": 185861 }
{ "_id": "QUEENS", "cun": 187471 }
{ "_id": "BROOKLYN", "cun": 221217 }
{ "_id": "", "cun": 256323 }
bye

olivier@olivier-HP-ZBook-15: ~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL mongo --port 3333 projetNoSQL < Water_Complaint_Street.js
MongoDB shell version: 2.6.10
connecting to: 127.0.0.1:3333/projetNoSQL
Classement des arrondissements en fonction du nombre de plaintes au sujet de l'eau

```

3.2.3 Agrégation des classements

Maintenant que l'ensemble des classements ont été effectué, on peut résumer les résultats obtenus sous la forme d'un tableau comme ci-dessous.

	Class _ Size	Graduation _ Outcomes	Hospitals _ Corporation _ Facilities	Landmarks	Motor _ Vehicle _ Collisions	Water _ Complaint _ Street	Wifi
Manhattan	1	2	2	1	3	3	1
Brooklyn	3	4	1	2	5	4	2
Queens	4	3	4	3	4	5	3
Bronx	2	5	3	4	2	2	4
Staten Island	5	1	5	5	1	1	5

TABLE 3.1 – Récapitulatif des résultats

A partir du classement des arrondissements on affecte un score de 1 à 5, 5 étant la meilleure note et 1 la moins bonne. Cela revient à inverser l'ordre du classement initial et la raison de cela sera expliquer au paragraphe suivant. On obtient donc le tableau suivant :

	Class_ Size	Graduation_ Outcomes	Hospitals_ Corporation_ Facilities	Landmarks	Motor_Vehicle_ Collisions	Water_ Complaint_ Street	Wifi
Manhattan	5	4	4	5	3	3	5
Brooklyn	3	2	5	4	1	2	4
Queens	2	3	2	3	2	1	3
Bronx	4	1	3	2	4	4	2
Staten Island	1	5	1	1	5	5	1

TABLE 3.2 – Tableau des scores des arrondissements

Il faut maintenant déterminer une manière d'agréger les résultats afin de dégager un classement global. Pour cela j'ai décidé d'utiliser une moyenne géométrique (ce qui explique l'inversion de l'ordre du classement exprimé dans le paragraphe précédent). L'utilisation d'une moyenne géométrique se justifie par la volonté de favoriser des arrondissements globalement bons dans de nombreux domaines plutôt que des arrondissements très bons dans des domaines mais également très mauvais dans d'autres. Par exemple un arrondissement étant troisième à deux reprises sera mieux classé qu'un arrondissement classé premier dans un classement et dernier dans un autre. ($\sqrt{3 \times 3} = 3 > \sqrt{5} = \sqrt{1 \times 5}$) On obtient finalement le classement suivant :

	Moyenne géométrique des scores
Manhattan	4,05
Brooklyn	2,67
Bronx	2,58
Queens	2,16
Staten Island	1,99

TABLE 3.3 – Classement final des arrondissements

Manhattan finit donc en tête du classement ! Bravo Manhattan !

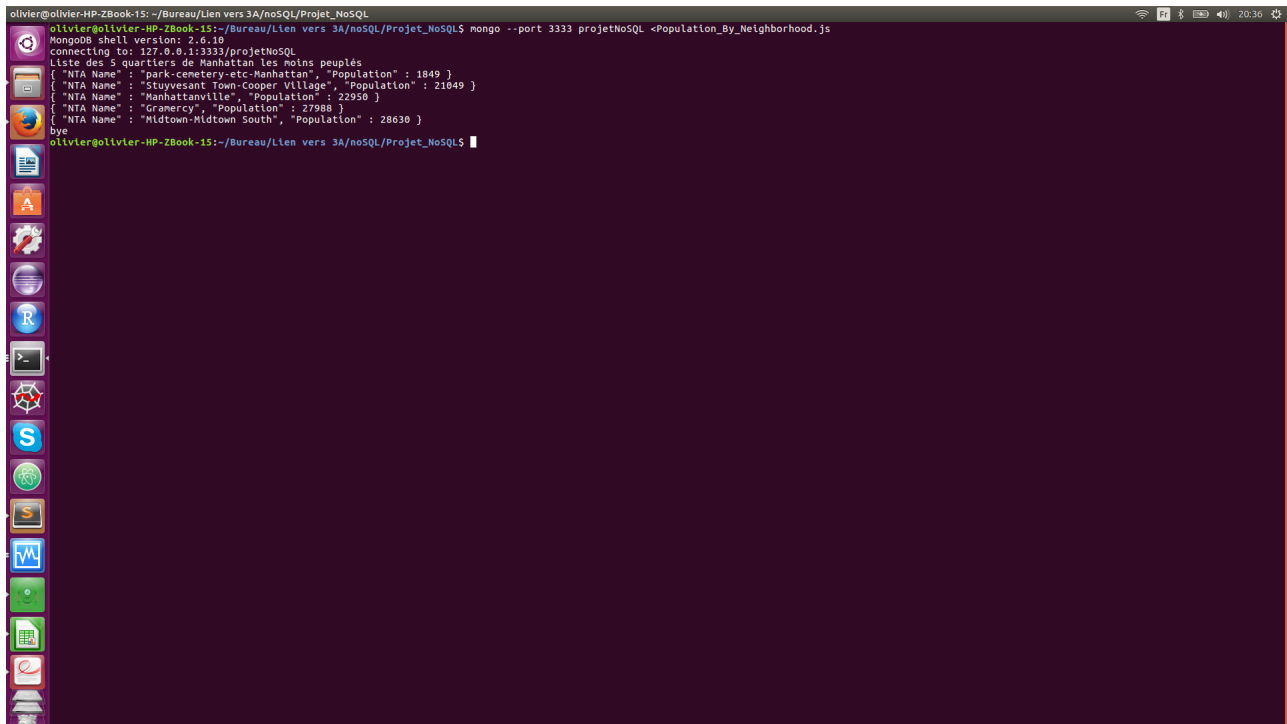
3.2.4 Et le gagnant est...

Maintenant que nous avons déterminé le meilleur des arrondissements selon de nombreux critères il nous faut maintenant déterminer un quartier intéressant. Pour déterminer ce quartier, on prendra comme critère le nombre d'habitant dans le quartier, en essayant de trouver le quartier le moins peuplé. En effet, si nous trouvons que Manhattan est le meilleur des arrondissements nous ne sommes sans doute pas les seuls à être arrivé à cette conclusion, il semble donc intéressant d'essayer de trouver un petit coin de sérénité au beau milieu de cette jungle urbaine.

En lançant dans un terminal la commande :

```
mongo -port 3333 projetNoSQL < Population_By_Neighborhood.js
```

On obtient donc les 5 quartiers les moins peuplés de Manhattan. Vous avez donc maintenant toute les clés en main afin de trouver l'appartement de vos rêves au sein de ces 5 quartiers de prédilection.



```
olivier@olivier-HP-ZBook-15: ~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL
olivier@olivier-HP-ZBook-15:~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL$ mongo --port 3333 projetNoSQL <Population_By_Neighborhood.js
MongoDB shell version: 2.6.10
connecting to: 127.0.0.1:3333/projetNoSQL
Liste des 5 quartiers de Manhattan les moins peuplés
{ "NTA Name" : "park-cemetery-etc-Manhattan", "Population" : 1849 }
{ "NTA Name" : "Stuyvesant Town-Cooper Village", "Population" : 21049 }
{ "NTA Name" : "Manhattanville", "Population" : 22950 }
{ "NTA Name" : "Gramercy", "Population" : 27988 }
{ "NTA Name" : "Midtown-Midtown South", "Population" : 28630 }
bye
olivier@olivier-HP-ZBook-15:~/Bureau/Lien vers 3A/noSQL/Projet_NoSQL$
```

PS : Si vous ne savez pas par où commencer vous pouvez lancer le script "Pour_bien_commencer" dans un terminal comme ceci :

bash Pour_bien_commencer

Chapitre 4

Discussion et axes d'améliorations

4.1 Le choix du type de base de données

Pour ce projet, j'ai décidé d'utiliser comme base de données une base MongoDB qui est une base de type document. Ce type de base paraissait adapté au vu du format des documents à stocker. Une base orientée graphe n'était clairement pas appropriée pour les données utilisées. L'utilisation d'une base orientée colonne ne semblait pas nécessaire. En effet, comme on ne prend les bases qu'à un instant t , il n'y a pas de problématique de mise à jour de base de données avec l'ajout de potentielles colonnes. Enfin, une base clé-valeur aurait pu être utilisée mais la base orientée document qui est en quelque sorte une extension des bases clé-valeur, semblait offrir une manipulation plus simple des données.

4.2 L'influence de la taille

Afin d'avoir une analyse plus représentative, on aurait également pu prendre en compte dans les analyses la taille des arrondissements. En effet, on peut s'attendre à ce qu'il y ait par exemple plus d'hôpitaux ou de monuments historiques dans les arrondissements plus grands. En récupérant des données à partir d'autres sources (Wikipédia par exemple), on aurait pu apporter une analyse plus représentative que celle qui a été menée précédemment.

4.3 L'utilisation des coordonnées spatiales.

Certaines bases de données comprenaient également des données spatiales telles que la latitude ou la longitude d'un monument par exemple. Cette information aurait pu être intéressante à prendre en compte mais elle apportait énormément de complexité quand à son traitement et aurait nécessité une réflexion plus longue sur la manière la plus appropriée de traiter ces données. En effet, dans ce type de situation, si l'on se pose la question de la distance à utiliser, une distance euclidienne n'aurait pas été appropriée et il aurait fallu se tourner vers d'autres types de distance tel qu'une distance de Manhattan par exemple. Or la prise en compte d'une telle distance aurait été particulièrement chronophage et je n'estimais pas que cela rentrait dans le spectre des compétences qui devait être évaluées au cours de cet enseignement.