

Projet
Passeport-Montréal V+
2018

BD6

Qiaoling He
Dominique Tessier

Table des matières

Objectif.....	3
Jeux de données.....	3
Requêtes.....	5
1. Le classement de stations en fonction d'utilisation.....	5
2. Les 5 plus importantes destinations pour chaque station.....	6
3. Déplacements par membre et non-membre.....	7
4. Tri du temps de déplacement.....	7
5. Déplacement par secteur.....	8
6. Déplacements entre les secteurs.....	8
7. Déplacements par « saut ».....	9
8. Impact du dénivelé sur les déplacements.....	9
9. Impact de la météo sur les déplacements.....	10
10. Proximité entre les stations BIXI et le métro.....	10
11. Proximité entre les stations BIXI et un centre d'intérêt.....	11
Annexe.....	11

Objectif

”BIXI est un système de vélo-partage à Montréal. Le réseau comprend 6250 vélos et 540 stations sur le territoire montréalais, ainsi qu’à Longueuil et Westmount. Beaucoup plus qu’un simple mode de transport, BIXI est aujourd’hui un fabuleux raccourci qui permet de circuler librement dans la ville où et quand on le désire pour aller où l’inspiration et/ou le devoir nous mène.”

Comprendre les déplacements avec BIXI des visiteurs et des montréalais permettrait d’offrir aux clients des mesures (temps de déplacement entre deux points) de plus en plus fiables. En utilisant les données ouvertes offertes par Bixi sur les déplacements et l’état de stations et en analysant les données désensibilisées, on pourrait savoir comment est utilisé BIXI :

- Quelles stations ou quels secteurs sont les plus utilisés
- Les déplacements entre les stations les plus courants

De plus, on pourrait *identifier les facteurs d’influence* comme

- L’impact de la météo sur les déplacements
- L’impact du dénivelé sur le déplacement
- Les caractéristiques des stations BIXI les plus utilisées?(ex. parcs, écoles, ... à proximité).
- La densité de la population est-elle un facteur essentiel pour choisir les prochaines stations?

Pour réaliser ce projet, les langages Python, MapReduce, Hive et Tableau seront utilisés.

Jeux de données

Cette section décrit les datasets utilisés pour faire les analyses.

1. Stations BIXI Historique des déplacements en BIXI
2. Stations STM
3. Météo
4. Densité de population de Montréal par secteur (à la recherche)

Stations BIXI	
Fichier source	Stations_2017.csv
Origin du fichier	https://www.bixi.com/fr/donnees-ouvertes
Dictionnaire de données	code: Identifiants de station utilisés correspondent à ceux de l'ensemble de données état des stations. name: Nom de station latitude: localisation horizontale de station longitude: localisation verticale de station
Exemple	7015,LaSalle / 4e avenue,45.43074022417498,-73.5919108253438

Historique des déplacements en BIXI	
Fichiers sources	OD_2017-04.csv: contient les déplacements du avril 2017 note: BIXI est accessible d’avril à novembre chaque année.
Origine du fichier	https://www.bixi.com/fr/donnees-ouvertes note: il y a les données d’avril 2014 à juin 2018.

Dictionnaire de données	start_date: Date et heure de début du déplacement au formant AAAA-MM-JJ hh:mm start_station_code: Identifiant de la station du début du trajet end_date ; Date et heure de la fin du déplacement au formant AAAA-MM-JJ hh:mm end_station_code : Identifiant de la station de fin du trajet is_member : Type d'utilisateur. Valeurs possibles 1 : Abonné du réseau BIXI Montréal, 0 : Non-abonné du réseau BIXI Montréal duration_sec: Durée total du déplacement en secondes
Exemple	2017-04-15 00:00,7060,2017-04-15 00:31,7060,1841,1
Méthodologie	Les trajets d'une durée de moins d'1 minute ou de plus de 2 heures sont exclus.

Station STM	
Fichiers sources	stops.txt
Origin du fichier	http://www.stm.info/fr/a-propos/developpeurs
Dictionnaire de données	stop_id (Numérique) : Identifiant unique de l'arrêt stop_code (Numérique) : Numéro client de l'arrêt stop_name (Texte variable): Nom de l'arrêt stop_lat (Numérique) : Latitude stop_lon (Numérique) : Longitude stop_url (Texte variable): Site de la stm (info arrêts) location_type: détermine si cet identifiant d'arrêt représente un arrêt, une station ou une entrée de station. 0 ou vide : arrêt, 1 : station, 2 : entrée/sortie de station parent_station: identifie la station associée à l'arrêt si celui-ci est situé au sein d'une station. wheelchair_boarding (Booléen): Accessibilité aux chaises roulantes pour cet arrêt. 1: accessible, 2: non accessible note: https://developers.google.com/transit/gtfs/reference/#stopstxt
Exemple	1,10280,Station Henri-Bourassa,45.555277,-73.668172,http://www.stm.info/metro/M01.htm,0,1S,1

Météo	
Fichiers sources	fre-hourly-04012017-04302017.csv
Origine du fichier	

Dictionnaire de données	Date/Heure, Année, Mois, Jour, Heure, Temp (°C), Temp Indicateur, Point de rosée (°C), Point de rosée Indicateur, Hum. rel (%), Hum. rel. Indicateur, Dir. du vent (10s deg), Dir. du vent Indicateur, Vit. du vent (km/h), Vit. du vent Indicateur, Visibilité (km), Visibilité Indicateur, Pression à la station (kPa), Pression à la station Indicateur, Hmdx, Hmdx Indicateur, Refroid. éolien, Refroid. éolien Indicateur, Temps
Exemple	"2017-04-30 23:00","2017","04","30","23:00","5,5","","3,7","","88","","5","","14","","24,1 ","","101,36","","","","","Pluie"

Requêtes

Pour répondre aux questions précédentes, on effectue des requêtes sur les jeux de données, soit en Hive, Pig ou MapReduce.

1. Le classement des stations en fonction de leur 'utilisation
2. Les 5 plus importantes destinations pour chaque station
3. Les déplacements par membres et non-membres
4. Les déplacements en fonction du temps
5. Les déplacements par secteur
6. Les déplacements entre les secteurs
7. Les déplacements par « saut »
8. L'impact du dénivelé sur les déplacements
9. L'impact de la météo sur les déplacements
10. La proximité entre les stations BIXI et le métro
11. La proximité entre les stations BIXI et un centre d'intérêt

1. Le classement des stations en fonction de leur² utilisation

Objectif : Déterminer les stations les plus achalandées et les destinations les plus importantes par « station de départ »

Programmation: Hive

Scripts:

```
-- Créer la table des stations BIXI
CREATE table BIXI_stations (
  station_id INT,station_name STRING,station_latitude float,station_longitude
float,altitude int,postalCode STRING)
PARTITIONED BY (year INT)
row format DELIMITED
FIELDS TERMINATED BY ' '
tblproperties("skip.header.line.count"="1");
```

```
-- Créer la table des déplacements en BIXI
CREATE table BIXI_OD (
  start_date timestamp, start_station_code int, end_date timestamp, end_station_code
  int, duration_sec int, is_member int)
PARTITIONED BY (year INT, month INT)
row format DELIMITED
FIELDS TERMINATED BY ',';

-- Les stations avec le plus d'arrivées
SELECT word, count(1) AS count FROM
(SELECT s.station_name AS word FROM BIXI_stations s
join BIXI_od od on s.station_id = od.end_station_code) w
GROUP BY word
ORDER BY count desc;
```

Output:

	word	count
0	Mackay / de Maisonneuve	2702
1	Berri / de Maisonneuve	2272
2	Métro Place-des-Arts (de Maisonneuve / de Bleury)	2046
3	Métro Mont-Royal (Rivard / du Mont-Royal)	1810
4	Métro St-Laurent (de Maisonneuve / St-Laurent)	1773
5	Métro Peel (de Maisonneuve / Stanley)	1765

```
-- Les stations avec le plus de départs
SELECT word, count(1) AS count FROM
(SELECT s.station_name AS word FROM BIXI_stations s
join BIXI_od od on s.station_id = od.start_station_code) w
GROUP BY word
ORDER BY count desc;
```

Output:

	word	count
0	Mackay / de Maisonneuve	2576
1	Métro Mont-Royal (Rivard / du Mont-Royal)	1835
2	Métro Laurier (Rivard / Laurier)	1812
3	Métro Peel (de Maisonneuve / Stanley)	1652
4	Métro Place-des-Arts (de Maisonneuve / de Bleury)	1627
5	du Mont-Royal / Clark	1566

2. Les 5 plus importantes destinations pour chaque station

Objectif: Identifier les destinations les plus importantes et compter les nombres de trajets entre la station départ et les stations arrivées.

Programmation: topEndStation2017.pig

Input: Historique des déplacements de 2017

Code:

```
deplacements = load '/pmv/deplacements/2017/*.*' USING PigStorage(',')
as(start_date,start_station_code,end_date,end_station_code,duration_sec,is_member);
deplacements_minus = FOREACH deplacements GENERATE
start_station_code,end_station_code,duration_sec;
start_end = GROUP deplacements_minus by (start_station_code,end_station_code);
start_end_freq = FOREACH start_end GENERATE flatten($0),COUNT($1) as count;
gStart_end_freq = GROUP start_end_freq by start_station_code;
topEndStation = foreach gStart_end_freq generate TOP(5,2,start_end_freq);
store topEndStation into '/pmv/topEndStation2017';
```

```

result_topEndStation2017_part-r-00000
{(5002,5006,34),(5002,5005,36),(5002,5004,41),(5002,5002,53),(5002,5007,122)}
{(5003,6034,21),(5003,5005,23),(5003,5006,39),(5003,5007,91),(5003,5004,31)}
{(5004,5005,24),(5004,5002,32),(5004,5004,33),(5004,5007,221),(5004,5003,44)}
{(5005,5003,20),(5005,5005,47),(5005,5004,27),(5005,5006,67),(5005,5007,225)}
{(5006,6116,46),(5006,5005,54),(5006,6501,50),(5006,5006,86),(5006,5007,275)}
{(5007,5007,134),(5007,5004,199),(5007,5002,182),(5007,5006,334),(5007,5005,312)}
{(6001,6114,192),(6001,6042,285),(6001,6052,212),(6001,6015,298),(6001,6043,304)}
{(6002,6015,124),(6002,6012,149),(6002,7020,132),(6002,6386,341),(6002,6114,311)}
{(6003,6032,121),(6003,6015,153),(6003,6064,168),(6003,6008,268),(6003,6100,196)}

```

Output: les 5 plus importantes destinations pour chaque station et leurs nombres de trajets.

Objectif: Pour vérifier le résultat de Pig du script précédent, on pourrait réaliser un MapReduce.

```

output_PMV_BIXI_topEndStations_2017
"5002" [[122, "5007"], [53, "5002"], [41, "5004"], [36, "5005"], [34, "5006"]]
"5003" [[91, "5007"], [39, "5006"], [31, "5004"], [23, "5005"], [21, "6034"]]
"5004" [[221, "5007"], [44, "5003"], [33, "5004"], [32, "5002"], [24, "5005"]]
"5005" [[225, "5007"], [67, "5006"], [47, "5005"], [27, "5004"], [20, "5003"]]
"5006" [[275, "5007"], [86, "5006"], [54, "5005"], [50, "6501"], [46, "6116"]]
"5007" [[334, "5006"], [312, "5005"], [199, "5004"], [182, "5002"], [134, "5007"]]
"6001" [[304, "6043"], [298, "6015"], [285, "6042"], [212, "6052"], [192, "6114"]]
"6002" [[341, "6386"], [311, "6114"], [149, "6012"], [132, "7020"], [124, "6015"]]
"6003" [[268, "6008"], [196, "6100"], [168, "6064"], [153, "6015"], [121, "6032"]]

```

Programmation: PMV_BIXI-topEndStations.py (MapReduce)

Input: Historique des déplacements de 2017

Output:

Note: les résultats des deux façons sont les mêmes.

Se référer à l'annexe pour une visualisation du résultat de cette requête.

3. Déplacements par membre et non-membre

Objectif : Identifier la part des membres dans les déplacements.

Programmation: PMV_BIXI_Member.py(MapReduce)

Input: Historique des déplacements de 2017

Output: le nombre de trajets par membres et non-membres en 2016 et 2017

output_PMV_BIXI_Member_2016	output_PMV_BIXI_Member_2017
"Member" 3230812	"Member" 3844806
"Non_Member" 769268	"Non_Member" 895551

4. Tri du temps de déplacement

Objectif : Identifier le nombre de déplacements par période de 15 minutes pour les membres et les non-membres.

Note : Des frais supplémentaires s'appliquent lorsque l'utilisateur dépasse le temps alloué pour son déplacement (30 minutes pour les non-membres et 45 minutes pour les membres).

Programmation: PMV_BIXI_Duration.py (MapReduce)

Input : Historique des déplacements de 2017

Output : le nombre de déplacements par période (et si membre (1) ou non (0))

output_PMV_BIXI_duration_2017	output_PMV_BIXI_duration_2016
[2, 0] 85054	[2, 0] 74105
[2, 1] 163289	[2, 1] 140912
[3, 0] 21717	[3, 0] 22183
[3, 1] 15340	[3, 1] 12793
[4, 0] 9596	[4, 0] 10377
[4, 1] 3155	[4, 1] 2562
[5, 0] 6044	[5, 0] 6473
[5, 1] 1326	[5, 1] 1058
[6, 0] 4231	[6, 0] 4650
[6, 1] 644	[6, 1] 519
[7, 0] 3187	[7, 0] 3484
[7, 1] 458	[7, 1] 381
[1, 0] 395842	[1, 0] 332112
[1, 1] 908733	[1, 1] 797594
[0, 0] 369880	[0, 0] 315884
[0, 1] 2751861	[0, 1] 2274993

Conclusion : Même si les membres sont plus nombreux, ce sont surtout les non-membres qui dépassent la limite imposée. Existe-t-il un forfait sans limite de temps?

5. Déplacement par secteur

Objectif : Déterminer le nombres de déplacements au départ et à l'arrivée pour chaque secteur (le code postal).

Programmation: PMV_BIXI-StartEndPostalCode.py

Paramètres : --BIXI_stations : Dataset sur les stations BIXI

Input : Historique des déplacements

Output : « Code postal » [Nombre de déplacements au départ, Nombre de déplacement à l'arrivée, différence]

```
H1M [21, 29, -8]
H1N [446, 527, -81]
H1T [559, 563, -4]
H1V [2331, 2652, -321]
H1W [3010, 3689, -679]
H1X [2054, 1854, 200]
...
```

On peut aussi avoir la même information mais par période. On a identifié 4 périodes en fonction (6-11 heures ; 11-16, 16-20 ; 20-6).

Output:

```
[H1M, 06-11] [5, 7, -2]
[H1M, 11-16] [4, 12, -8]
[H1M, 16-20] [8, 7, 1]
[H1M, 20-06] [4, 3, 1]
[H1N, 06-11] [103, 116, -13]
[H1N, 11-16] [160, 174, -14]
[H1N, 16-20] [142, 185, -43]
[H1N, 20-06] [41, 52, -11]
[H1T, 06-11] [89, 148, -59]
...
```

Note : Il faudrait mettre en option mais pour l'instant, ce n'est qu'un commentaire qu'on met ou non.

Déplacements en fonction de quatre périodes identifiées sans égard à la localisation par code postal.

Output:

```
[06-11] [38971, 38971, 0]
```



```
[11-16] [60060, 60060, 0]
[16-20] [65565, 65565, 0]
[20-06] [31731, 31731, 0]
...
```

Note : Idem au résultat précédent mais la clé sur le code postal est mise en commentaire

6. Déplacements entre les secteurs

Objectif : En fonction du code postal, déterminer le nombre de déplacements entre les secteurs.

Programmation: PMV_BIXI-StartEndKey.py (MapReduce)

Paramètres :

```
--BIXI_stations : Dataset sur les stations BIXI
--PostalCodeLength : Nombre de caractères à conserver pour le code postal pour faire le
regroupement des stations
--output-dir « local directory » --no-output : rediriger le résultats vers un fichier qui servira
d'input au programme PMV_Sankey.py
```

Input : Dataset des déplacements en BIXI

Output : [CodePostal départ, Code postal arrivée], nombre de déplacements

```
[H1, H1] 6330
[H1, H2] 5437
[H1, H3] 485
[H1, H4] 25
[H1, H5] 5
[H1, J4] 4
[H2, H1] 6258
[H2, H2] 103022
...
```

Note: Référez à l'annexe pour une visualisation du résultat de cette requête.

7. Déplacements par « saut »

Un usager qui dépasse son temps alloué (30 minutes pour les non-membres et 45 minutes pour les membres) pour se déplacer d'une station à une autre paye des frais supplémentaires. Il peut cependant faire autant d'étapes sans frais supplémentaire.

Objectif : Trouver la station intermédiaire entre deux stations pour le faire en moins de temps.

Programmation: Hive

Exemple: Trouver la station intermédiaire entre les stations « Métro Langelier (Sherbrooke / Langelier) » (7016) et « Island / Centre » (6350) pour la plus petite durée de déplacement.

```
select s1.start_station_code, s1.end_station_code, s1.duration_sec,
s2.start_station_code, s2.end_station_code, s2.duration_sec,
(s1.duration_sec + s2.duration_sec) as duration
from BIXI_startendduration s1
```

	s1.start_station_code	s1.end_station_code	s1.duration_sec	s2.start_station_code	s2.end_station_code	s2.duration_sec	duration
1	7016	6113	1769	6113	6350	1314	3083

```
join BIXI_startendduration s2 on s1.end_station_code = s2.start_station_code
where
s1.start_station_code = 7016 and
s2.end_station_code = 6350
order by (s1.duration_sec + s2.duration_sec)
Limit 1;
```

Output:

On pourrait savoir que la meilleure station intermédiaire est la 6113 (Alexandre-De Sève / de Maisonneuve) en une cinquantaine de minutes.

Note : Basé sur l'historique des déplacements, on a bâti une table qui contient la durée moyenne des déplacements entre deux stations,

```
create table BIXI_StartEndDuration (
start_station_code int, end_station_code int, duration_sec int);
```

```
insert into BIXI_StartEndDuration
select start_station_code, end_station_code, round(avg(duration_sec))
from BIXI_od
group by start_station_code, end_station_code;
```

8. Impact du dénivelé sur les déplacements

Objectif : Calculer le nombre de déplacements d'une station à une autre lorsque le dénivelé est différent (plus haute ou plus basse) et calculer la moyenne du dénivelé entre les stations pour l'ensemble des déplacements.

Programmation: Hive

```
-- Nombre de déplacements vers une station plus basse ou plus haute
select count(od.start_date) as cnt
from BIXI_od od
left join BIXI_stations b1 on od.start_station_code = b1.station_id
left join BIXI_stations b2 on od.end_station_code = b2.station_id
where (b1.altitude - b2.altitude) < 0 ; -- vers une station plus haute
-- where (b1.altitude - b2.altitude) > 0 ; -- vers une station plus basse
```

Output :

- Vers une station plus basse : 2 277 774
- Vers une station plus haute : 1 761 325

```
-- moyenne du dénivelé entre les stations pour tous les déplacements (par mois d'utilisation)
select month, count(od.start_date) as cnt,
avg(b1.altitude - b2.altitude) as avg from BIXI_od od
left join BIXI_stations b1 on od.start_station_code = b1.station_id
left join BIXI_stations b2 on od.end_station_code = b2.station_id
group by month;
```

Output:

	month	cnt	avg
1	4	196327	3.2278851100459947
2	5	196326	3.2279015515010747
3	6	748078	3.1702322485088454
4	7	877338	3.0775003476425278
5	8	859471	3.1331993749643674
6	9	747633	3.3858564295583529
7	10	569930	3.6040338287158074
8	11	151728	3.7914755351681957

Conclusion : Les utilisateurs utilisent leur vélo dans une direction et reviennent ou partent en utilisant un autre moyen de transport.

9. Impact de la météo sur les déplacements

Objectif : Déterminer le nombre de déplacements en fonction du temps ou de la vitesse du vent.

Programmation: PMV_BIXIMeteo.py (MapReduce)

Paramètres :

- Meteo : Dataset sur les enregistrements météo
- Mode : Extraire suivant le temps (1) ou la vitesse du vent (2)

Input : Dataset des déplacements en BIXI

Note :

Le lien entre les datasets se fait sur la date (yyyymmddhh)

Le Dataset pris est le premier de la saison et ne compte que 16 jours. Il faudrait faire le traitement sur un autre mois (au mieux sur toute l'année)

Output: Résultats suivant le temps

Averses de pluie	[7629, 15, 508.60]
Brouillard	[5382, 15, 358.80]
Bruine	[1665, 4, 416.25]
Bruine,Brouillard	[1051, 9, 116.78]
Dégage	[11917, 40, 297.93]

Généralement degage	[59873, 63, 950.37]
Généralement nuageux	[59932, 92, 651.43]
Granules de glace ou gresil	[708, 1, 708.00]
Nuageux	[30334, 75, 404.45]
Pluie moderee	[128, 1, 128.00]
Pluie moderee,Brouillard	[153, 2, 76.50]
Pluie	[14864, 51, 291.45]
Pluie,Brouillard	[2691, 16, 168.19]

Output: Résultats suivant la vitesse du vent

Moins de 10 km/h	[16587, 57, 291.00]
Moins de 20 km/h	[67193, 164, 409.71]
Moins de 30 km/h	[67553, 117, 577.38]
Moins de 40 km/h	[34285, 38, 902.24]
Moins de 50 km/h	[9025, 7, 1289.29]
Moins de 60 km/h	[1684, 1, 1684.00]

10. Proximité entre les stations BIXI et le métro

Objectif : Trouver les stations BIXI à proximité des stations de metro de la STM (moins de 100 mètres). Le résultat sera inséré dans une table et pourra servir à d'autres requêtes.

Programmation: Hive

Script:

```
CREATE TABLE BIXI_stm_Distance100 AS
Select distinct b.station_id, b.station_name, s.stop_id, s.stop_name,
Distance2 (s.stop_lat, s.stop_lon, b.station_latitude, b.station_longitude) as distance
from BIXI_stations b, stm_stations s
where Distance2 (s.stop_lat, s.stop_lon, b.station_latitude, b.station_longitude) < 100;
```

Output: Un échantillon de la requête (juste le Select)

	b.station_id	b.station_name	s.stop_id	s.stop_name
0	6001	Hôtel-de-Ville 2 (du Champs-de-Mars / Gosford)	12-02	Station Champ-de-Mars - Édicule St-Antoine
1	6001	Hôtel-de-Ville 2 (du Champs-de-Mars / Gosford)	124654	Station Champ-de-Mars (St-Antoine / Gosford)
2	6004	Hôtel-de-Ville (du Champs-de-Mars / Gosford)	12-02	Station Champ-de-Mars - Édicule St-Antoine
3	6004	Hôtel-de-Ville (du Champs-de-Mars / Gosford)	124654	Station Champ-de-Mars (St-Antoine / Gosford)
4	6009	Ste-Catherine / Labelle	11-01	Station Berri-UQAM - Édicule Sainte-Catherine
5	6009	Ste-Catherine / Labelle	118673	Station Berri-UQAM (Berri / Ste-Catherine)
6	6009	Ste-Catherine / Labelle	118676	Station Berri-UQAM (Berri / Place Émilie-Gamelin)

11. Proximité entre les stations BIXI et un centre d'intérêt

Objectif : Trouver les 3 stations BIXI les plus près d'un centre d'intérêt en se servant de sa position (longitude et latitude). Dans l'exemple suivant on utilisera le centre d'intérêt suivant à la latitude :45.540301 et à la longitude : -73.602041.

Programmation : Hive

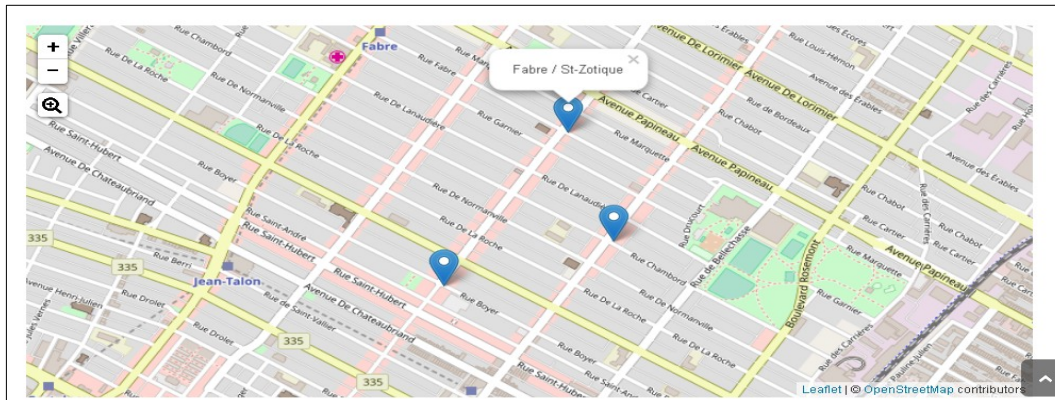
Script:

```
select distinct
b.station_id, b.station_name,
Distance2 (45.540301, -73.602041, b.station_latitude, b.station_longitude) as distance
from BIXI_stations b
order by distance limit 3;
```

Output:

	b.station_id	b.station_name	distance
0	6268	Chambord / Beaubien	214.3787140575766
1	6255	Boyer / St-Zotique	341.233370300664
2	6270	Fabre / St-Zotique	359.93711071043526

- Même genre de requête mais avec la longitude, latitude pour afficher sur une carte.
select distinct b.station_id, b.station_latitude, b.station_longitude, b.station_name, Distance2
(45.540301, -73.602041, b.station_latitude, b.station_longitude) as distance
from BIXI_stations b
order by distance limit 3;



Annexe

1. Visualisation du résultat de la requête 2: les 5 plus importantes destinations pour chaque station.

Programmation: Tableau

Input:

Stations_2017.csv: information sur les stations BIXI

Abc les5PlusImport...	Abc les5PlusImporta...	Abc les5PlusImport...	📍 les5PlusImporta...	📍 les5PlusImportant...	# les5PlusImporta...
Type	Code	Path	Latitude	Longitude	Nombre
Origin	7015	7015_6425	45,4307402	-73,591911	303
Origin	7015	7015_6715	45,4307402	-73,591911	259
Origin	7015	7015_7057	45,4307402	-73,591911	213
Origin	7015	7015_7048	45,4307402	-73,591911	172
Origin	7015	7015_7058	45,4307402	-73,591911	138
Destination	7015	6715_7015	45,4307402	-73,591911	444
Destination	7015	6712_7015	45,4307402	-73,591911	278
Destination	7015	6714_7015	45,4307402	-73,591911	166

Les5PlusImportantDestinations.csv : généré par topEndStations2017.py avec le fichier output de requête 2 et Stations_2017.csv

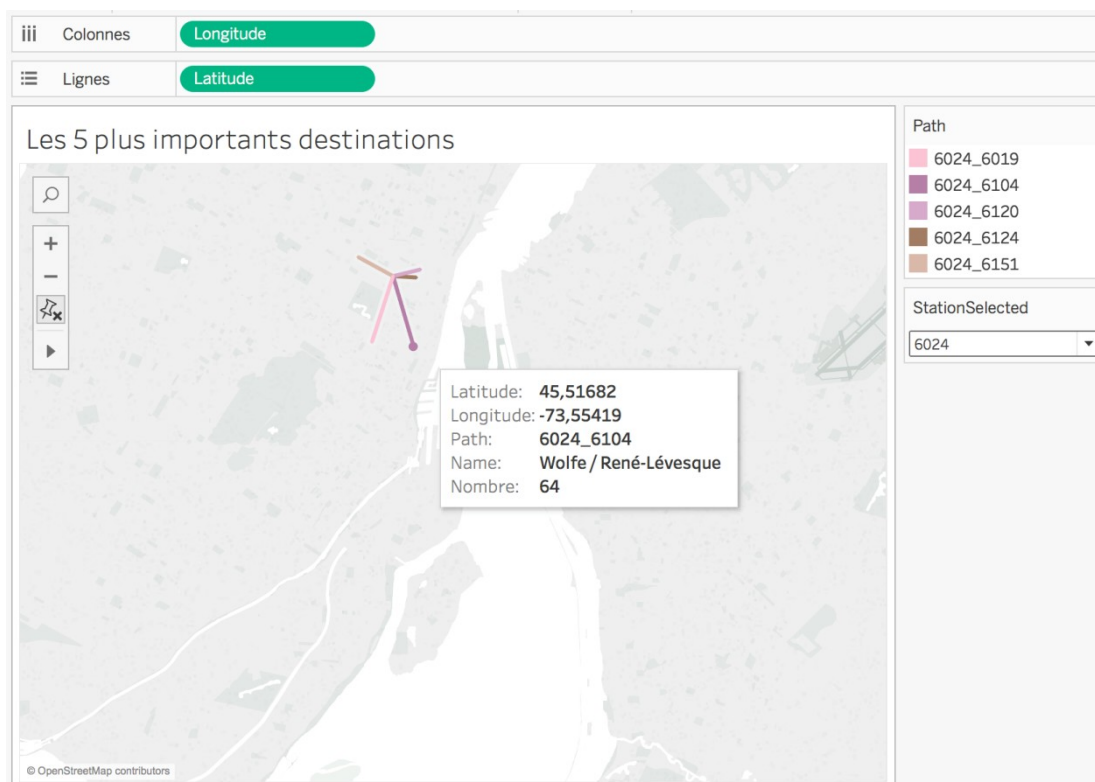


Illustration 1: Une carte de Montréal avec 5 lignes indiquant les liens entre les stations.

2. Visualisation du résultat de la requête 6: Déplacements entre les secteurs

Programmation: PMV_Sankey.py

Input : Fichiers générés par le programme PMV_BIXI-StartEndKey.py.

Dessin 1: Diagramme de type « sandkey » pour montrer les déplacements entre les secteurs.

