Étude de cas:

Analyse de marché du transport aérien canadien avec R

Atelier d'introduction à R

Beauchemin, David

Cabral Cruz, Samuel

Goulet, Vincent

Dans le cadre du colloque R à Québec

25 mai 2017

Table des matières

Préface

Dans le cadre du colloque "R à Québec" qui se tiendra le 25 et 26 mai 2017 sur le campus de l'Université Laval, une séance d'introduction au langage de programmation R sera offerte aux participants. Cette séance vise principalement la compréhension et la pratique permettant de maîtriser les rudiments de cet environnement de programmation. [3] Cette séance sera divisée en deux parties. En ce qui concerne la première partie, les fondements du langage seront visités d'une manière théorique sous la forme d'un exposé magistral. La deuxième partie, tant qu'à elle, se concentrera davantage sur la mise en pratique des notions abordées lors de la première partie grâce à la complétion d'une étude de cas cherchant à faire l'analyse de marché du transport aérien canadien. Ce document correspond en fait à la documentation complète de cette deuxième partie de formation.

Étant donné qu'il s'agit tout de même d'une formation pour débutants, la majorité du code sera déjà fournie, mais il n'en vaut pas moins la peine de parcourir ce projet si ce n'est que pour constater la puissance et la simplicité du langage. De plus, il est souvent difficile de mettre en perspective les innombrables fonctionnalités d'un langage lorsque nous commençons à l'utiliser. Cet étude de cas nous fournit ainsi un bel exemple d'enchaînement de traitements jusqu'à l'aboutissement ultime qui consiste à répondre aux questions que nous nous posions avant même d'amorcer l'analyse.

D'autre part, il est important de préciser que le code qui sera présenté ne correspond pas toujours à la manière la plus efficiente d'accomplir une tâche donnée. L'objectif principal étant ici la transmission de connaissances dans un dessin éducatif plutôt que d'une réelle analyse de marché. Nous tenons aussi à mentionner que bien qu'il s'agisse d'une formation s'adressant à des débutants, plusieurs notions qui seront mises en valeur font plutôt état de niveau intermédiaire et avancé, mais apportées toujours de manière simplifiée et accessible à quiconque qui n'aurait jamais travaillé avec R.

Nous tenons à remercier Vincent Goulet de nous avoir fait confiance dans l'élaboration de cette partie de la formation ainsi que tous les membres du comité organisationnel de l'évènement. Nous croyons sincèrement que R est un langage d'actualité qui se révèle un atout à quiconque oeuvrant dans un domaine relié de près ou de loin aux mathématiques.

Introduction

Dans le cadre de cette étude de cas, nous nous placerons dans la peau d'un analyste du département de la tarification oeuvrant au sein d'une compagnie canadienne se spécilisant dans le transport de colis par voies aériennes en mettant à profit le jeu de données d'*OpenFlights*. [2]

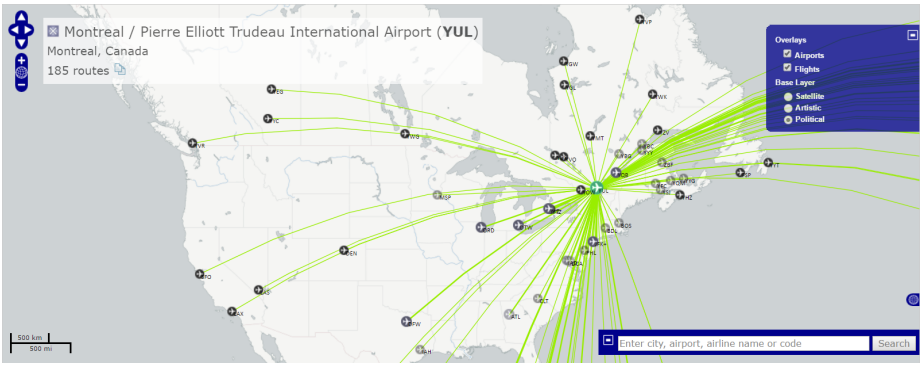

FIGURE 1 – Interface de l'outil OpenFlights

Parmi les bases de données disponibles, nous retrouvons :

- airports.dat - les données relatives à tous les aéroports du monde
- routes.dat - les données relatives à tous les trajets possibles dans le monde
- airlines.dat - les données relatives à toutes les compagnies aériennes du monde

Ainsi, notre mandat consistera, dans un premier temps, à analyser les bases de données mises à notre disposition afin de créer des fonctions utilitaires qui permettront de facilement intégrer les informations qu'elles contiennent lors de la tarification d'une livraison spécifique. Une fois cette tarification complétée, nous devrons fournir des chartes pour facilement estimer les prix d'une livraison qui s'avèreront être des outils indispensables au département de marketing et au reste de la direction. Après avoir transmis les documents en question, votre gestionnaire voulant s'assurer que la nouvelle tarification sera efficiente et profitable vous demandera d'analyser les prix de la concurrence pour en extrapoler

leur tarification. Finalement, vous serez appelé à comparer ces deux tarifications et la compétitivité de votre nouvelle tarification comparativement au reste du marché en procédant à une analyse stochastique.

OpenFlights

OpenFlights est un outil en ligne permettant de visualiser, chercher et filtrer tous les vols aériens dans le monde. Il s'agit d'un projet libre entretenu par la communauté via GitHub. [1] L'information rendu disponible y est étonnament très complète et facile d'approche ce qui en rend ce jeu de données très intéressant pour quiconque qui désire s'initié à l'analyse statistique.
https://openflights.org/

Bien qu'on n'en soit toujours qu'à l'introduction, nous tenons dès lors introduire des notions de programmation qui comparativement à celles qui suivront sont d'autre un peu plus général. Tout d'abord, afin de maximiser la portabilité des scripts que vous créerez dans le futur, il est important de rendre votre environnement de travail indépendant de la structure de dossier dans laquelle il se trouve. Pour ce faire, nous devons donc utiliser le principe de liens relatifs plutôt qu'absolus. En R, deux fonctions bien spécifiques nous fournient les outils afin de rendre cette tâche possible. Il s'agit de *getwd()* et *setwd()*. Comme leurs noms l'indiquent, elles servent respectivement à extraire le chemin vers l'environnement de travail et à le modifier.

De manière similaire qu'au sein d'un invité de commandes traditionnel, il est possible d'utiliser ".." (cd ..) afin de revenir à un niveau supérieur dans la structure de dossiers. Dans la plupart des cas, le code source d'un projet sera souvent isolé du reste du projet en le plaçant dans un sous-dossier dédié. (Il s'agit ici d'une excellente pratique de programmation et je dirais même indispensable si vous utilisez un gestionnaire de versions.)

Bref, comme le code source du présent projet se retrouve à l'intérieur du sous-dossier *dev* et que nous pourrions vouloir avoir accès à d'autres parties du répertoire au sein du code, le code suivant nous permettra de placer notre racine de projet à un niveau de dossier supérieur et de stocker ce chemin dans la variable *path*. Avec cette variable, tous les appels subséquents à des portions du répertoire pourront donc ce faire de manière relative puisque c'est cette variable path qui changera d'une architecture à un autre, tandis que la structure du répertoire restera toujours la même.

La deuxième notion que nous tenons à introduire immédiatement est celle de reproductibilité d'une analyse statistique. Comme vous le savez probablement, l'aléatoire pur n'existe pas en informatique, d'où la raison pour laquelle nous utiliserons plutôt le terme de nombres pseudo-aléatoires. Bien que cela peut sembler étrange à première vue, il existe tout de même un point positif à tout

ceci, soit la possibilité de fixer une racine au générateur de nombre pseudo-aléatoire (GNPA) ce qui aura comme impact de toujours produire les mêmes résultats pour autant que le GNPA utilisé soit le même. Comme nous pouvons le voir dans le code ci-dessous, l'instruction *set.seed()* se chargera de fournir une valeur de départ aux calculs du GNPA.

```
#### Setting working directory properly ####
setwd('..')
path <- getwd()
set.seed(31459)
```

Étude de cas

1.1 Extraction, traitement, visualisation et analyse des données

HELLO

1.2 Création de fonctions utilitaires

1.3 Communication des résultats

1.4 Analyse de la compétition

1.5 Ajustement de distribution statistiques sur données empiriques

1.6 Simulation et analyse de rentabilité

Conclusion

Bibliographie

[1] Github.

[2] Openflights.

[3] R à québec 2017.