Document technique pour tableau de bord MOOC IMT

Ecrit par Giovanni Zanitti – 04/08/2016

1. **Objectif du document**

Ce document a pour objectif d’expliquer comment fonctionne l’application en partant de l’outil utilisé jusqu’au déploiement sur un serveur interne de l’application en passant par tous les calculs d’indicateur. Cependant, le code étant déjà commenté, je ne rentrerais pas trop dans le détail dans ce document. De plus, le code est assez redondant une fois qu’on a compris comment était structuré les fichiers ainsi que la façon de faire des graphiques, c’est pourquoi certaines parties sont moins commentées que d’autres. De plus, ces commentaires ont seulement pour but d’expliquer la structure des fichiers et non pas de vous apprendre les commandes du logiciel R ainsi que de ses packages.

1. **Le logiciel R et le package R-shiny**

R est un logiciel libre de traitement des données et d’analyses statistiques mettant en œuvre le langage R. Il fait partie des logiciels statistiques les plus connus et par conséquent possède une très grande communauté qui est très active sur les forums d’aide. Si R dispose dans sa version de base la plupart des fonctionnalités utiles pour la statistique courante, ses possibilités s’élargissent dès que l’on utilise les paquets, souvent écrits en R et mis librement à disposition.

J’ai donc utilisé cette possibilité pour établir mon tableau de bord en utilisant le Package R-Shiny qui permet de développer des tableaux de bords dynamiques.

Ce package propose une structure bien particulière pour développer un tableau de bord avec notamment deux fichiers :

* Server.R : fichier contenant tout le code qui sert à calculer les indicateurs et à faire les graphiques.
* UI.R : fichier qui contient tout ce qui concerne l’interface utilisateur (« UI » pour  « User Interface »).

Le lien entre ces deux fichiers se fait à l’aide d’une fonction prenant dans deux listes « Input » et « output » aussi bien des graphiques que des données. Ceci permet notamment de faire une interface dynamique avec des listes déroulantes ou des zones de saisies qui font directement réagir les indicateurs affichés.

Pour plus de clarté dans mes fichiers, j’ai utilisé dans server.R la fonction « source » de R qui permet de faire appel à un script d’un autre fichier pour pouvoir écrire du code (notamment l’importation des données) dans un autre fichier que j’ai appelé « Analyse.R ».

1. **Récupération des données**

La récupération est fastidieuse du fait de l’absence d’une base de données. Pour obtenir les données que l’on souhaite incorporer dans le tableau de bord il faut posséder un compte « Enseignant » sur le MOOC en question et aller récupérer les données dans le « fun/dashboard » au format csv. Une fois ce fichier téléchargé, il faut tout de suite supprimer la première ligne (contenant le titre du fichier) qui ne sert à rien. De plus, on remarque que les noms des attributs associés aux questions ne sont pas explicites (« q1 », « q2 », « q3 », …). De plus, le codage des réponses n’est pas plus parlant (« choice\_0 »,  « choice\_1 », …). C’est pourquoi, j’ai dû mettre en place un processus de transformation des données pour que celles-ci soient compréhensibles et toutes du même format.

Dans le dossier « Pré-traitement » (qui se trouve dans les documents de la machine virtuelle), vous trouverez donc les données brutes rangées par MOOC et par session ainsi que les fichiers de réponses. Ces derniers ont été écrit à la main et reprennent les intitulés des réponses aux questions. Pour importer de nouveaux MOOC, il faut placer les fichiers dans un nouveau dossier au nom du MOOC et suivant la même structure que les fichiers déjà créés. Une fois ce fichier créé, il faut compléter la liste du fichier questions.csv dans le dossier Application si une variable jamais vue avant a été insérée (par exemple si une question est bien spécifique à un MOOC, il faut la rajouter dans la liste).

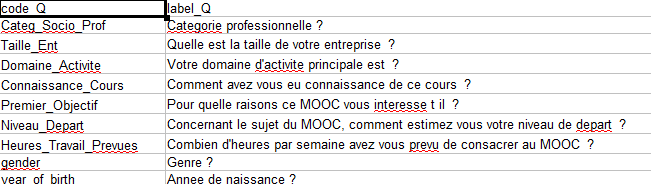


Figure 1 - Extrait de questions.csv

La première colonne étant le nom de la variable dans le fichier des réponses et la deuxième l’intitulé exact de la question. **ATTENTION** : L’application ne supporte pas les accents, il faut donc faire en sorte de ne pas en mettre dans le fichier.

Une fois ces étapes réalisées, ouvrez le fichier « Transformation2.R » (les autres fichiers étant des tests en tout genre), renseignez les champs demandés et lancez le programme. Les fichiers traités se trouveront dans le dossier « Application/data/MOOC/ » suivi du nom du MOOC ainsi que de la session.

1. **Structure du code**

Dans le dossier « Application » se situant dans le dossier « srv/shiny-server » (dans la machine virtuelle), vous trouverez donc les données comme dis précédent mais également les fichiers faisant tourner l’application.

* Le fichier « server.R » est structuré de manière à avoir les objets utilisés partout en haut puis tous les codes de calculs d’indicateurs onglets pas onglets. Chacun de ces codes sont commentés.
* Le fichier « UI.R » contient d’abord les informations relatives à la barre grisée située à gauche de l’écran puis contient toutes les fonctions servant à faire fonctionner les onglets.
* Enfin le fichier « analyse.R » contient les imports de données ainsi que le traitement de certaines variables.

A noter qu’il faut avoir les droits super-admin pour modifier ces fichiers puisqu’ils sont dans le dossier srv. De plus, pour les éditer, j’ai utilisé Rstudio qui est également installé sur la machine virtuelle.

1. **Lancement de l’application**

L’application est donc stockée sur un server interne de R-shiny sur une machine virtuelle Ubuntu. Pour faire cette installation j’ai suivi ce tutoriel : <http://www.developpez.net/forums/blogs/71025-bouye/b719/installer-configurer-serveur-r-shiny-ubuntu/>. Ce tutoriel fonctionne plutôt bien dans l’ensemble, mis à part des paquets à installer manuellement suite à des erreurs données dans le terminale. Par exemple, j’ai dû installer « libssl-dev\_1.0.1 » ou encore « libcurl4-openssl-dev » tout seul mais cela dépend de la machine qu’on a à disposition.

Conformément à ce tutoriel, il ne vous reste plus qu’à lancer le server grâce à la commande suivante : sudo service shiny-server start, et à aller dans le navigateur en tapant http://[numéro IP de la machine]/Application.