



Travail pratique

1 Mise en situation


Une jeune entreprise en démarrage veut commercialiser auprès des clubs de golf une application permettant de prévoir le total de la ronde d'un quatuor en cours de partie. Les clubs utiliseraient l'application dans le cadre des tournois organisés sur leur terrain à la fois comme un jeu — la prévision s'avérera-t-elle exacte? — et comme un défi additionnel — le quatuor saura-t-il battre la prévision?

L'entreprise fait appel à votre expertise pour concevoir le modèle mathématique de prévision et un prototype d'application.

De votre côté, vous reconnaissez immédiatement dans ce problème de prévision des résultats d'un quatuor à partir de ses résultats passés et de ceux d'autres quatuors un contexte d'hétérogénéité tout à fait propice à l'application des modèles de théorie de la crédibilité.¹


2 Mandat

Votre mandat comporte deux volets principaux. Le premier consiste à élaborer un modèle utilisant la théorie de la crédibilité apte à prévoir le total d'une ronde de golf d'un quatuor à partir de ses pointages aux trous précédents, le nombre de ceux-ci pouvant aller de un à dix-sept. Le mandataire exige d'élaborer au moins trois modèles de nature différente², de les comparer et de justifier votre choix final.

Dans le second volet du mandat, vous devez concevoir et mettre en production un prototype d'application de calcul des prévisions à l'aide de la technologie [Shiny](#) .


3 Données disponibles

La configuration et le niveau de difficulté varient beaucoup d'un terrain de golf à l'autre. Dans ce contexte, il devient essentiel de calibrer l'applica-

1. Ce travail est une adaptation, réalisée par Frédéric Guillot puis Vincent Goulet, d'un problème proposé à l'origine par Réseau Action TI Montréal et SAS lors de l'édition 2015 de la compétition d'idées ouverte [Actulab](#) .

2. Trois modèles bayésiens avec des paramètres différents ne font donc pas le compte.

tion de prévision pour le terrain où elle sera utilisée.


Aux fins de développement de votre modèle et du prototype d'application, vous aurez accès à des données provenant du Club de golf Beaconsfield. Celles-ci se présentent sous la forme de trois fichiers en format **CSV**  utilisant le symbole « ; » pour délimiter les colonnes ³.

resultats-complets.csv Résultats complets de 2 000 rondes de golf. La colonne `Game_ID` contient un identifiant unique pour une ronde; les colonnes `Score_hole_1` à `Score_hole_18` contiennent les résultats (en nombre de coups) obtenus par un quatuor au trou correspondant.

resultats-partiels.csv Résultats partiels de 600 rondes de golf assimilable à des parties en cours. Les colonnes `Game_ID` et `Score_hole_1` à `Score_hole_18` sont identiques à celles du fichier précédent. Un résultat vide pour un trou signifie que le quatuor n'a pas encore joué ce trou. La colonne `TOTAL` servira à inscrire vos prévisions pour ces rondes.

normales.csv Normales (*par*) des 18 trous du Club de golf Beaconsfield.

4 Informations additionnelles

Les tournois de golf se déroulent habituellement selon la formule « **quatre balles, meilleure balle**  » (ou *Vegas*, ou *Continuous Mulligan*), ce qui veut essentiellement dire que le pointage d'une ronde est celui d'un quatuor, et non d'un joueur individuel.

De plus, les départs des quatuors durant un tournoi de 18 trous s'effectuent de manière simultanée, chacun à un trou différent du parcours (*shot-gun*). Ainsi, un quatuor qui débute, par exemple, au trou numéro 8 complètera sa ronde dans l'ordre suivant : 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1, 2, 3, 4, 5, 6, 7.

L'application de calcul des prévisions sera utilisée par une équipe volante qui rencontrera chaque quatuor à tour de rôle durant un tournoi, sans ordre préétabli. Par conséquent, tant le nombre que la liste des trous complétés varieront d'un quatuor à l'autre. Si l'équipe de prévision rencontrait le quatuor mentionné ci-dessus au trou numéro 1, seuls seraient disponibles les résultats des trous 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 et 18.

L'équipe de prévision ne sera pas en mesure d'enregistrer les résultats des quatuors au fil du tournoi; elle ne pourra qu'entrer les résultats d'un quatuor dans l'application et fournir la prévision.

3. Vous pouvez importer ces données dans R directement avec la fonction `read.csv2`.

Vous aurez l'occasion de rencontrer les mandataires du projet afin de préciser leurs attentes et leurs besoins au regard du prototype d'application. Consultez le plan de cours pour les modalités.

5 Livrables

Les livrables du mandat sont principalement constitués de code source R Markdown et Shiny, ainsi que de fichiers de données. Ce sont tous des fichiers en format **texte brut** [↗](#). Lorsqu'ils contiennent du texte avec des accents, ils doivent être enregistrés dans le codage de caractères **UTF-8** [↗](#).⁴

Les sections ci-dessous détaillent les différents livrables. Ils doivent se trouver dans la branche principale (**master** ou **main**) du dépôt Git précisé dans le plan de cours. Sauf avis contraire, les livrables doivent se trouver dans le répertoire racine du dépôt.

Consultez le plan de cours pour les dates et les autres modalités de livraison.



Pour être considéré livré, un fichier de code source doit permettre de générer l'objet qu'il définit (fichier PDF, application) dans un environnement R et standard sans intervention manuelle.

5.1 Rapports d'étape

Les rapports d'étape font état de vos progrès dans le mandat, en plus de proposer une analyse critique de vos méthodes de travail en équipe. La longueur des rapports d'étape est limitée à **une page**.

Tous les rapports d'étape doivent obligatoirement aborder les points suivants :

1. Mandat (ce que nous avons à accomplir);
2. Réalisations (ce que nous avons accompli);
3. Apprentissages (ce que nous avons appris).

Le livrable du **premier rapport d'étape** est un fichier `rapport-etape-1.Rmd` contenant le code source R Markdown du rapport. Outre les points obligatoires ci-dessus, le rapport d'étape doit expliquer, en quelques phrases, les mesures que vous comptez mettre en place dans la suite afin d'améliorer la productivité de votre équipe.


4. L'UTF-8 n'étant pas le codage par défaut sous Windows, portez une attention toute particulière à ce point si vous utilisez ce système d'exploitation.

Le livrable du **second rapport d'étape** est un fichier `rapport-etape-2.Rmd` contenant le code source R Markdown du rapport. Outre les points obligatoires ci-dessus, le rapport d'étape doit comparer le temps que vous avez effectivement consacré à la seconde partie du mandat à l'estimation que vous aurez faite dans le cadre d'un Café du savoir. Expliquez les écarts. Avez-vous manqué de temps et, si oui, pourquoi ?

Le livrable du **troisième rapport d'étape** est un fichier `rapport-etape-3.Rmd` contenant le code source R Markdown du rapport. Outre les points obligatoires ci-dessus, le rapport d'étape doit expliquer brièvement l'organisation et le partage des tâches au sein de votre équipe. Avez-vous apporté des changements à ce chapitre depuis la présentation du premier rapport d'étape ?

5.2 Livraison de fin de mandat

La livraison de fin de mandat compte un rapport final, vos résultats de prévision et un prototype d'application mis en production. En détail :

1. Un fichier `rapport-final.Rmd` contenant le code source R Markdown d'un rapport final d'**au plus trois pages** faisant état de votre choix de modèle de prévision et de vos conclusions. Le rapport final doit aborder les points obligatoires des rapports d'étape, en plus d'expliquer : ce que vous avez amélioré dans votre travail du mandat ; ce que vous avez trouvé difficile dans ce mandat et pourquoi.
2. Vos prévisions pour les 600 rondes du fichier `resultats-partiels.csv`. Celles-ci permettront aux mandataires de juger de la qualité de votre modèle à l'aide d'un test statistique objectif. Vous devez rendre le fichier *dans le même format que vous l'avez reçu* avec vos prévisions inscrites dans la colonne TOTAL.
3. Le code source de votre prototype d'application dans un répertoire `prototype`. La structure de celui-ci doit permettre de seulement fournir le nom du répertoire à la fonction `shiny::runApp` pour lancer l'application. Vous devez documenter le code source de l'application de manière appropriée.
4. Votre prototype d'application accessible et opérationnel depuis un serveur d'applications Shiny. Le plus simple consiste à publier votre application sur shinyapps.io , un service d'hébergement gratuit (en partie) offert par RStudio. Insérez l'url vers votre application de manière bien visible dans le fichier `README.md` de votre dépôt.



Nous devons pouvoir importer vos prévisions dans R avec la commande `read.csv2` sans autre argument que le nom du fichier.

6 Remarques finales

- Il faut jouer au moins un coup pour compléter un trou au golf.
- Au-delà des modèles de crédibilité étudiés dans le cours, le modèle hiérarchique de Jewell (1975) et le modèle de régression de Hachemeister (1975) pourraient s'avérer dignes d'intérêt pour ce projet. Consultez Bühlmann et Gisler (2005) et la vignette `credibility` du paquetage **actuar** (Dutang et collab., 2008) pour les détails sur ces modèles et l'évaluation numérique des primes de crédibilité.

Bibliographie

- Bühlmann, H. et A. Gisler. 2005, *A Course in Credibility Theory and its Applications*, Springer, ISBN 3-5402575-3-5.
- Dutang, C., V. Goulet et M. Pigeon. 2008, «actuar: An R package for actuarial science», *Journal of Statistical Software*, vol. 25, n° 7. URL <http://www.jstatsoft.org/v25/i07>.
- Hachemeister, C. A. 1975, «Credibility for regression models with application to trend», dans *Credibility, theory and applications*, Proceedings of the Berkeley actuarial research conference on credibility, Academic Press, New York, p. 129-163.
- Jewell, W. S. 1975, «The use of collateral data in credibility theory: A hierarchical model», *Giornale dell'Istituto Italiano degli Attuari*, vol. 38, p. 1-16.

