

# R Visualisation - Projet

## Football Tracking Dashboard

*A. Gardahaut - G. Tsang - T. Brient*

*11 mars 2019*

## Présentation

L'application Shiny porte sur des données sur le Football (joueurs, clubs, transferts). Elle est composée de trois onglets thématiques :

- L'onglet **Players** qui affiche la liste des joueurs, avec quelques caractéristiques importantes. Il est donné à l'utilisateur la possibilité de filtrer cette base à partir de nombreux paramètres (scores, club, individuels, positions, etc.). En cliquant sur un joueur dans la table principale, une fiche individuelle du joueur apparaît à l'écran de l'utilisateur. D'ici, le joueur est davantage détaillé, notamment à travers de nombreux scores mais également à partir de deux représentations graphiques. Le premier permet d'avoir un aperçu rapide du profil du joueur et de pouvoir le comparer avec un autre joueur. Attention ! Le graphique radar n'est pas le même selon qu'on visualise un joueur de terrain (c'est-à-dire hors gardien) et un gardien. Donc la liste des joueurs à comparer est conditionnée au joueur étudié. Le second graphique est la projection des scores de performance selon la position sur le terrain. Les gardiens n'ayant pas vocation à s'éloigner de leurs cages, ils n'ont pas de représentation de ce type.
- L'onglet **Clubs network** qui affiche, pour un club donné (laissé au choix de l'utilisateur), les différents clubs avec lesquels il a effectué au moins un transfert (entrant ou sortant). La longueur des flux des réseaux est négativement proportionnelle au prix de la transaction. Ainsi, les clubs proches du centre (logo du club sélectionné initialement) ont réalisés de gros transferts avec ce club. Des séries temporelles sont tracées sur le côté droit de l'onglet, représentant respectivement, le budget lors des mercatos (Achats moins Ventes), le nombre total de transferts (entrées + sorties), le nombre d'entrées et de sorties, tous, par saison (une saison équivalant à deux Mercatos).
- L'onglet **Transfers map** qui affiche, selon le mode choisi, les flux géographiques des transferts. L'utilisateur peut choisir de voir les transferts d'un joueur (étude de son évolution de carrière), les transferts d'un club (étude des partenaires) ou des transferts effectués pour une ou plusieurs saisons de Mercatos (étude des activités). Dans tous les cas, l'utilisateur a la possibilité de placer son curseur sur les flux et d'avoir des informations supplémentaires sur ces derniers. Des graphiques additionnels sont accessibles via les boutons rouges placés en haut de la carte, permettant de lire facilement l'évolution de la valeur d'échange d'un joueur et de visualiser les plus grosses transactions effectuées selon les critères de sélection appliqués.

## Objectifs

En marge de vouloir aboutir sur une application fonctionnelle, permettant à l'utilisateur intéressé de rechercher et d'observer des informations relatives aux joueurs et à l'activité économique des principales ligues mondiales à l'heure actuelle et l'historique depuis la saison 2000-2001; une grande partie du travail était dédiée à la finition et au rendu de l'interface. Les *inputs* sont généralement choisis et ordonnés de façon à rendre l'ergonomie de l'application meilleure. Par ailleurs, quelques modifications CSS de l'application ont été apportées pour améliorer l'esthétique des onglets **Clubs network** et **Transfers map**. L'onglet **Transfers map** étant très chargée en informations et en zones de paramètres, un travail fin sur le placement et les dimensions des différents éléments était indispensable (utilisation de panels transparents, limitation en taille des menus déroulants, etc.).

On espère que tous les cas de figure obtenus par les choix de l'utilisateur ont été couverts et traités de façon à ne pas le laisser dans la confusion (messages d'information en cas de restrictions trop fortes dans la carte leaflet par exemple). Les logos de clubs non disponibles dans la base de données sont remplacés par des icônes par défaut si besoin. Un ballon de football a été placé dans le coin supérieur, à droite de l'application. Celui-ci est en mouvement lorsque la machine est en procédure de calculs afin d'informer l'utilisateur.

Enfin, le challenge d'étaler des flux superposés dans la partie cartographie a été relevé par Axel (fonction `fct_trace` dont les explications sont en fin de document). Tout a été pensé de manière à ce que les bases de données utilisées puissent être mises à jour et augmentées sans avoir à modifier le code en conséquence. Il nous semble tout à fait envisageable d'adapter ce projet comme un outil de visualisation et d'étude à des fins marketing ou financières.

## Construction des bases de données

Les bases de données ont été récupérées sur le site [kaggle.com](https://www.kaggle.com). Des modifications en amont ont été réalisées (modification des unités du système Américain en système International par exemple) et de nouvelles variables ont été rajoutées. Ces nouvelles variables concernent l'emplacement géographique des clubs. Par webscrapping sur des requêtes successives à partir des noms de clubs, la ville et pays des clubs ont pu être obtenus. Puis par l'utilisation de la fonction `geocodeGratuit`, ont été extraites les coordonnées associées. Les clubs dont le processus n'a pas été concluant ont été traités manuellement.

Lorsque plusieurs clubs se situent dans la même ville (56 villes concernées), un léger décalage dans l'espace (grâce à un tirage aléatoire) a été effectué pour éviter la superposition des logos sur la carte.

## Limites et bugs connus

### Limites

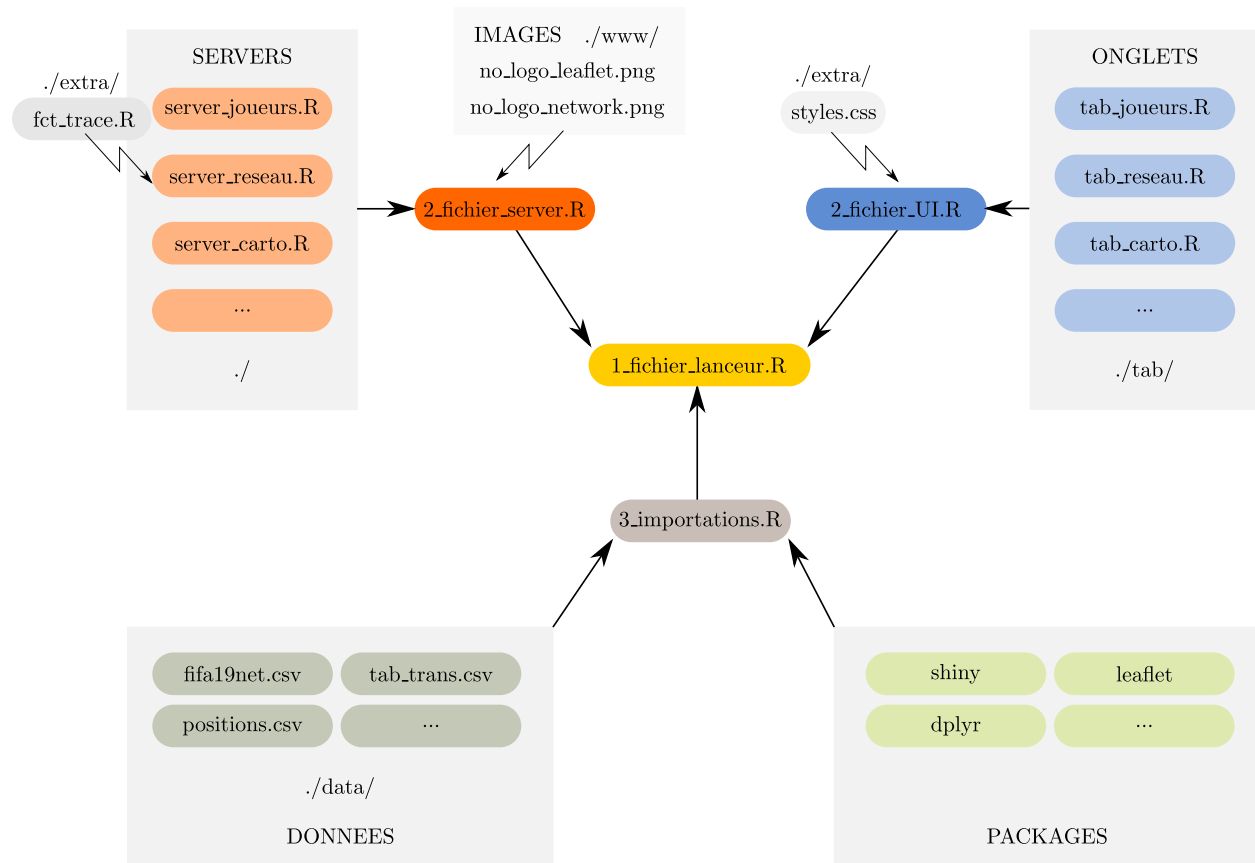
- Les médias, notamment les fichiers images des photos des joueurs, des logos de clubs et des drapeaux de pays sont obtenus à partir d'urls extérieurs ([sofifa.com](https://www.sofifa.com)). Ceci peut ralentir le bon déroulement de l'application (temps de téléchargement des images).
- Par manque de temps, il manque la possibilité d'interagir entre les différents onglets (faire apparaître une fiche joueur complète (celle de l'onglet **Players**) lorsqu'on étudie ses transferts dans l'onglet **Transfers map** par exemple).
- Après étude du temps d'exécution de l'application à partir de l'option profilage de Rstudio, l'importation du fichier `bg_terrain.rds` (fond de graphique en forme de terrain de football pour la fiche-joueur) nécessite entre 1.5 et 2 secondes. Ce fichier `rds` issu d'un `rasterGrob` sur une image `png` pourrait être évité en utilisant un autre moyen d'afficher une image en fond de graphique *ggplot*.
- ...

### Bugs connus

- Les `knobInput()` (similaires à des `sliderInput()` mais sous forme de cercle) présents dans le menu déroulant dans les paramètres de filtrage des joueurs dans l'onglet **Players** n'apparaissent pas. Il est demandé à l'utilisateur d'effectuer la manipulation suivante : (1) ouvrir le menu déroulant contenant ces `knobInput()` et (2) modifier la taille de la fenêtre de l'application. Ils seront alors affichés et fonctionnels.
- Une quantité non négligeable de Clubs n'ont pas de logos. Ceci s'explique par l'absence d'urls dans la base de données de départ d'une part, et d'autre part, de noms de Clubs différents selon la base de donnée utilisée. Un `fuzzyjoin` ainsi qu'une vérification manuelle ont été effectués. Des images par défaut ont été utilisées en cas d'absence, permettant de compléter la base en flux continu.

## Organisation du répertoire

Le programme de l'application Shiny est découpé en plusieurs morceaux à des fins de lisibilité et d'ergonomie de développement. La figure suivante présente le fonctionnement et la structure du répertoire dans sa globalité.

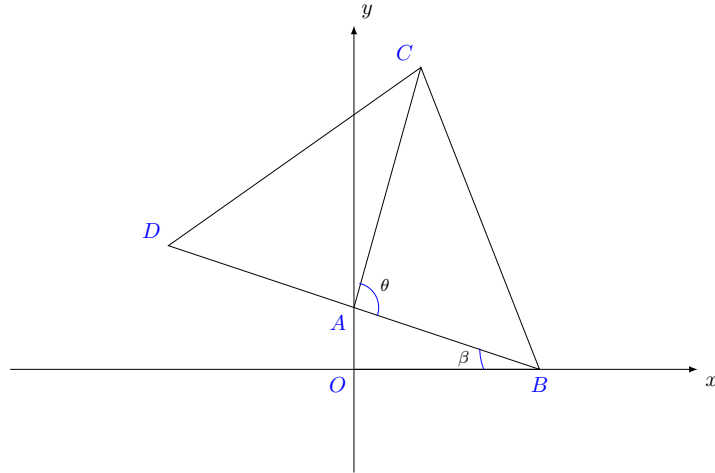


## Documentation de la fonction N\_flows\_0D (A. Gardahaut)

Dans certains cas, nous souhaitons tracer plusieurs flux entre paires de points (Clubs pour nous), hors les fonctions habituellement utilisées en R et Leaflet ne permettent pas de générer plusieurs flows distincts, en conséquence, ils se superposent.

Nous souhaitons donc résoudre ce problème.

L'idée est basée sur le constat suivant, nous pouvons visualiser les N flux entre une même paire de points si ils sont tous de cambrures différentes.



La cambrure étant déterminée par un angle  $\alpha \in [0^\circ, 90^\circ]$ , avec :

$$\alpha = \widehat{ABC}$$

L'idée pour générer les flux entre B et D est d'interpoler  $n$  points entre B et D via le polynôme de Lagrange  $P_A(x^*)$  associé aux 3 points B, D et A, milieu de B et D dans le repère  $(x^*, y^*)$  formé respectivement par (BD) et son perpendiculaire passant par B, une fois ce repère recentré en  $(0, 0)$  (afin d'utiliser les repères d'Euler). On prendra autant de valeurs de  $\alpha$  que de flux voulus.

On recentre d'après le point (B ou D) dont l'abscisse est la plus proche de  $(0)$ .

$$P_A(x^*) = \Delta \frac{(x^* - x_B^*)(x^* - x_D^*)}{(x_A^* - x_B^*)(x_A^* - x_D^*)}$$

Avec :

$$\Delta = \sin(\alpha) \frac{d(B, D)}{2}$$

**Remarque 1** Pour éviter d'avoir un  $\beta > 90^\circ$ , on change l'orientation du vecteur si nécessaire (c'est-à-dire si  $\vec{BD} = (a, b)$ ,  $a < 0$ ).

Ensuite on calcule :

$$(\cos(\beta), \sin(\beta)) = \left( \frac{\langle \vec{BD}, (1, 0) \rangle}{\|\vec{BD}\|}, \frac{(1, 0) \wedge \vec{BD}}{\|\vec{BD}\|} \right)$$

On calcule les coordonnées des points B, D, A dans  $(x^*, y^*)$ , on subdivise l'intervalle  $[x_B^*, x_D^*]$  en  $n$  points  $(x_i)_{i \in [1:n]}$  et on calcule les :  $P_A(x_i^*)$  associés.

Le retour dans la base  $(x, y)$  se fait via :

$$\begin{aligned} \vec{x} &= \cos(\beta)x^* - \sin(\beta)y^* \\ \vec{y} &= \sin(\beta)x^* + \cos(\beta)y^* \end{aligned}$$

Pour N pair, on distribue les flux  $j$  et  $j + 1$  selon  $\pm \alpha$  puis on augmente de  $\alpha$  pour la prochaine paire de flux.

$$\alpha_j = \lfloor i/2 \rfloor (-\alpha)^j \quad j \in [2, N]$$

On prend par défaut  $\alpha = 10^\circ$ .

Pour le cas impair, on initialise le premier flux comme étant le segment passant par les 2 points (c'est-à-dire si  $\alpha = 0^\circ$ ), puis pour les autres, on procède comme dans le cas pair.