

## DATA 732 TP2

Le but de ce TP était d'apprendre à se servir des différents outils pour récupérer, trier et représenter des données disponibles sur python

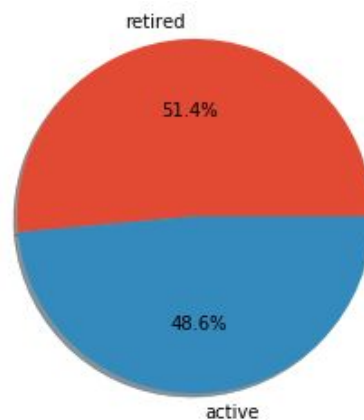
Le code de notre projet se trouve ici: <https://github.com/Yan420/DATA-732-TP2>

### Partie 1:

#### Répartition de l'activité des coaches:

Ce graphique se fait grâce à la fonction `types_coaches(df)`.

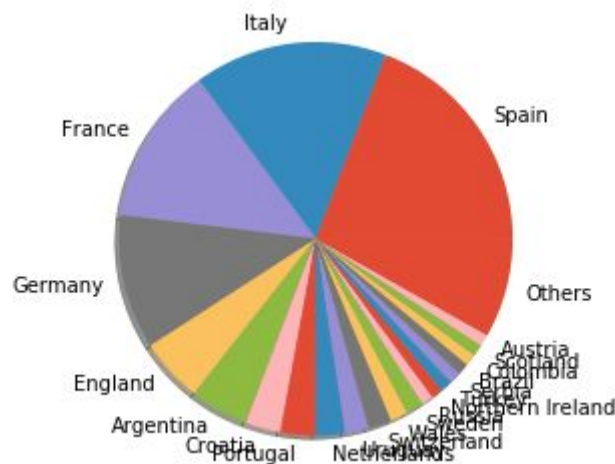
Le programme va d'abord regarder combien de lignes le data Frame comprends. Sachant qu'il y a une ligne par coachs, le nombre de lignes du data Frame représente le nombre totale de coaches. Le programme va ensuite regrouper les coaches par "currentTeamId", car les coaches qui ne s'occupent pas d'une équipe ont un 0 dans cette colonne. Avec cela on connaît le nombre total de coaches et le nombre total de coaches qui ne s'occupent pas d'équipes, on peut alors tracer le graphique sous matplotlib.



### Répartition des coaches par pays:

Ce graphique se fait grâce à la fonction `coaches_per_country(df)`.

Ici, la colonne importante est la colonne "passportArea" car elle contient entre autre le nom du pays où le coach agit. On regroupe les lignes par "passportArea" et on obtiens alors le nombre de coaches par pays. Il suffit de mettre ces données dans deux listes pour un graphique et utiliser matplotlib.



Les pays n'ayant qu'un seul coach sont dans la catégorie "Others" pour éviter d'avoir un graphique illisible

### Placement des équipes sur une carte:

Cette carte se fait grâce à la fonction `map_teams(df_teams)`.

Cette fonction va tout d'abord retirer toutes les équipes de type "club" du data Frame car nous voulons les équipes nationales. Pour tous les noms de villes restants, nous allons utiliser le module "geolocator" qui est une API qui prends un nom en argument et renvoie ses coordonnées sur une carte. L'API ne trouvant pas tout le temps, quand elle ne trouve pas un nom, elle affiche ce nom dans la console. On va ensuite se servir de Basemap pour afficher une carte de la terre avec un point pour toutes les villes trouvées.

La carte:



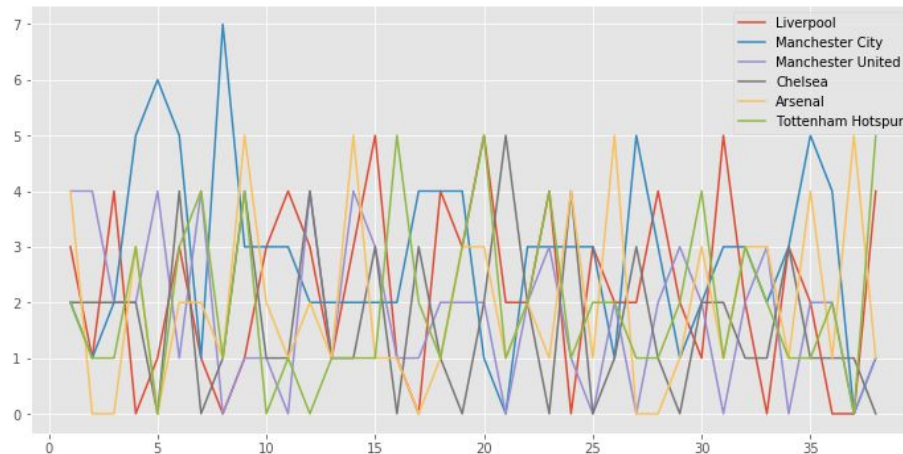
Les villes non trouvées:

```
Tiran\u00eb (Tirana)
Bucure\u015fti
Bogot\u00e1 D.C.
Ciudad de Panam\u00e1
Br\u00f8ndby
San Jos\u00e9
T\u014dky\u014d (Tokyo)
Tehr\u0101n (Teheran)
Reykjav\u00eak
Coyoac\u00e1n, Ciudad de M\u00e9xico (D.F.)
Ar-Riy\u0101\u01e0d (Riyadh)
```

### Graphique buts Big 6:

Ce graphique se fait grâce à la fonction `curves_big6(df_teams, df_matches_en)`.

Premièrement, ce programme va chercher l'id des 6 équipes en question dans `df_teams`. Ensuite, pour chacun des matchs, le programme va regarder si au moins l'une des 6 équipes joue. Si c'est le cas, le programme va regarder le numéro de la semaine du match et recorder la valeur du nombre de but dans la liste des buts de l'équipe. Une fois que ceci est fait pour tout les matchs, le programme crée le graphique.



### Emplacement buts terrain:

Cette représentation se fait grâce à la fonction `goals_team_field("Liverpool", df_teams, df_events_en)`. (pour Liverpool)

Premièrement, le programme va chercher l'id de l'équipe choisie. Ensuite le programme va regarder les événements un à un et va retirer ceux qui ne sont pas fait par l'équipe désignée ou qui ne sont pas des "Goal kick". On se retrouve avec un data Frame de tous les buts de l'équipe désignée. Pour chacun de ces événements, on va récupérer la valeur "positions". L'un des deux position étant celle du but, on peut l'ignorer. On a donc une liste de positions que l'on va placer sur une représentation de terrain de football grâce au module `mplsoccer`. Ce programme prend près de 3 minutes à s'exécuter à cause de la très grande quantité de données.

