CentraleSupélec

Soutenance Finale

Stimul x Latitudes

4 juin 2018

Stimul

latitudes
exploring tech
for good_

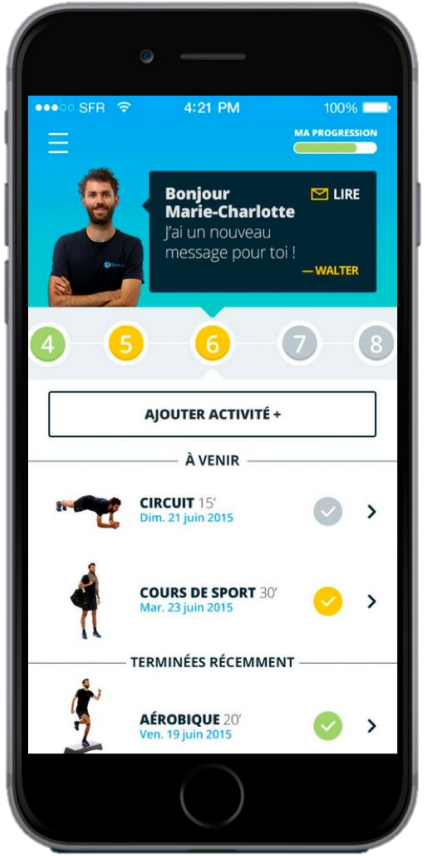
Plan

I. Présentation de Stimul
II. Rétrospective sur la phase d’exploration
III. Résultats de l’analyse des données
IV. Suite donnée au projet
V. Conclusion

I. Présentation de Stimul

Programme de thérapie digitale dans le cadre de maladies chroniques :

- Meilleure qualité de vie (alimentation & activité physique)
- Réduire les effets/risques de ces maladies
- Suivi par des coachs spécialisés
- Utilisation d'objets connectés



II. Quels problèmes et enjeux ?

➔ **<u>Enjeu</u> :** analyse de données de **194** users pour dégager des tendances, des corrélations sur ce jeu de données à partir de **KPI** identifiés en début de projet

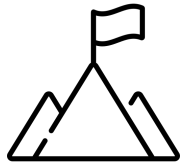
Leçons réalisées
Perte de poids
METs (*Metabolic Equivalent of Task*)
Nombre de pas
Interaction avec l'interface

➔ Mise en place de méthodes générales qui seraient applicables quelque soit le jeu de données

II. Retour sur les risques identifiés en phase d'exploration

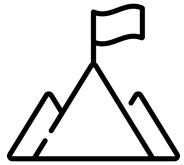
01	**Nombre de données : fiabilité de la tendance (santé)**	• Globalement peu de points de donnée donc des résultats à prendre dans leur contexte et avec recul Mise en place de méthodes générales réutilisables
02	**Fiabilité de certaines données**	• Tri et nettoyage des données incohérentes, et extrapolation à partir des moyennes si besoin
03	**Utilisation des données textes issues des conversations.**	• Nous n'avons pas traité cette partie

III. Synthèse des avancées du projet (½)



Appropriation et extraction des données	Elaboration du code pour KPI	Matrice de corrélation	KPI d'interaction
Récolte et familiarisation avec tous les différents tableaux et fichiers .csv fournis par Stimul et première approche des données.	Construction du code pour dégager et calculer les KPIs (pas, MET, poids et leçons) de chaque utilisateur disponible.	Mise en forme et calcul de la matrice de corrélation pour visualiser l'influence des différents facteurs	Définition et calcul d'un nouveau KPI des interaction que les utilisateurs ont avec la plateforme Stimul.

III. Synthèse des avancées du projet (2/2)



Temporalité par cohorte	Interactions et .json	Jupyter Notebook	Sémantique et dashboarding
Analyse de l'évolution des KPI par cohorte. Introduction de la notion de temporalité par cohorte	Exploration des fichiers .json de Google Analytics et mise en forme de données exploitables pour obtenir des informations sur l'implication de l'utilisateur	Mise en forme de tout notre code de manière à le rendre lisible, et exploitable simplement pour un nouveau projet	Ouverture et potentiel futur projet

III. Livrable : 3 Jupyter Notebook

Calcul des KPI

Implication sur les 3 premières semaines

On va compter le nombre d'actions volontaires effectuées par chaque user. Ceci exclue par exemple les nombres de pas qui sont enregis podomètres, et qui necessitent juste d'avoir l'application ouverte. On commence par ecrire une fonction qui compte les actions volontaire metric durant les 21 premiers jours. La fonction prend en argument le tableau metrics, et un tableau time début qui contient les dates de d pour chaque user.

Pour faire ceci, on créé une liste c qui va sélectionner les user_id dans metrics (il faut ensuite supprimer les doublons). On créé une liste le nomnbre d'actions faites par chaque user. Ensuite, pour chaque user, on créé un tableau A pour chaque user, où on sélectionne les ligr des actions volontaires. On se réfère au tableau activity_category pour cela. Ensuite on créé un tableau B qui ne contient en fait qu'une va du programme pour le user en question. Ensuite on compte les actions effectuées moins de 21 jours après le début du programme.

```
def count_voluntary_actions(metric,time_debut):

    c = np.array(metric['user_id'])
    c = np.array(list(set(c)))
    tot = []

    for i in range(len(c)):
        A = metric[metric['user_id'].isin([c[i]])]
        A = A[A['category_id'].isin([14,15,16,17,18,19,20,21,22])]
        B = time_debut[time_debut['user_id'].isin([c[i]])]

        A = A.reset_index(drop = True)
        B = B.reset_index(drop = True)

        n,l = 0,A.shape[0]

        for k in range(0,l) :
            if A.loc[k,'date']-B.loc[0,'createdAt'] <= timedelta(days=21):
                n+=1
        tot.append(n)
```

Calcul de l'évolution du nombre de pas en fonction du temps

Tout d'abord on commence par creer un nouveau DataFrame qui va contenir toutes les lignes où les utlisateurs marchent ou courent (category_id vaut 5, 6 ou 7). Ensuite, on enlève la colonne des categories, puisqu'elle nous est inutile, et on enlève tous les doublons. Enfin, on fait une jointure entre la table contenant les pas et les dates des differentes activités, et la table contenant le numero de cohorte de chaque utilisateur. La jointure se fait donc sur l'attribut commun du "user_id". On fait un "outer merge" pour que les utilisateurs présents dans les deux tableaux soient présents dans le tableau final. Ceci a pour effet de prendre en compte les utilisateurs qui auraient été affectés à une cohorte, mais qui n'auraient pas fourni de données sur le nombre de pas qu'ils ont effectués.

```
pas = metric[metric['category_id'].isin([5,6,7])]
pas = pas.drop('category_id', axis = 1)
pas = pas.drop_duplicates()

df_merged = pd.merge(pas ,cohorte, how = 'outer', on = 'user_id')
```

Traitement des fichiers JSON. ¶

```
import pandas as pd
import json
from matplotlib.pyplot import *
from pandas.io.json import json_normalize
import dateparser
import numpy as np
import datetime
from datetime import timedelta
```

On commence par écrire des fonctions intermédaires qui vont permettre de traiter les fichiers json. La première permet d'importer un fichier json dans python.

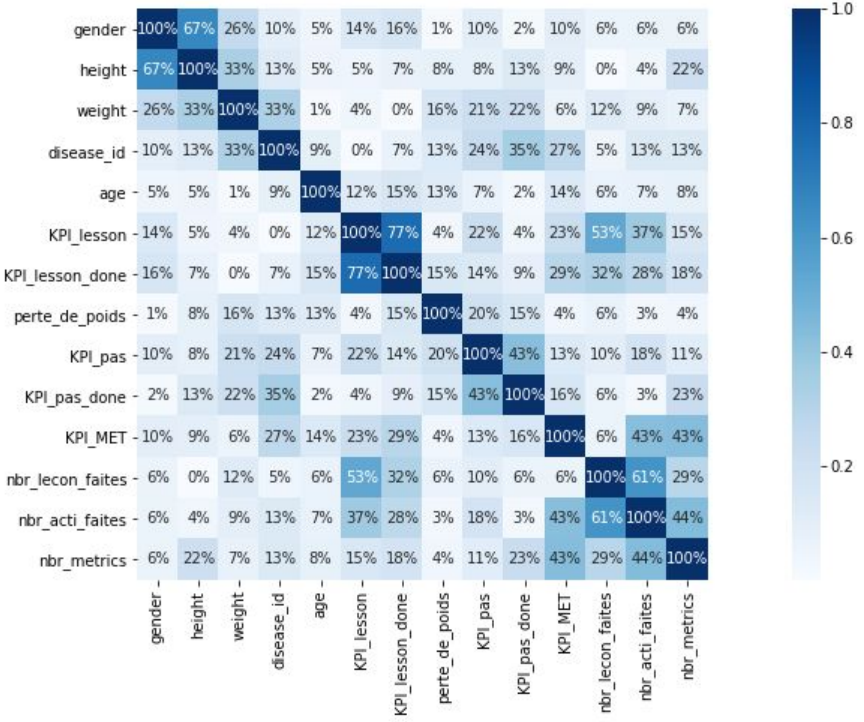
```
def importation(file):
    with open(file) as json_data:
        d = json.load(json_data)
    return(d)
```

La fonction minute prend en argumment un string de la forme 'hh:ss' et renvoie le nombre de secondes corespondant. Cette fonction va permettre de convertir les colonnes 'duration' des fichiers importés.

```
def minutes(str_heure) :
    duree = str_heure.split(':')
    m, s = int(duree[0]), int(duree[1])
    t = timedelta(minutes=m, seconds=s)

    return(t)
```

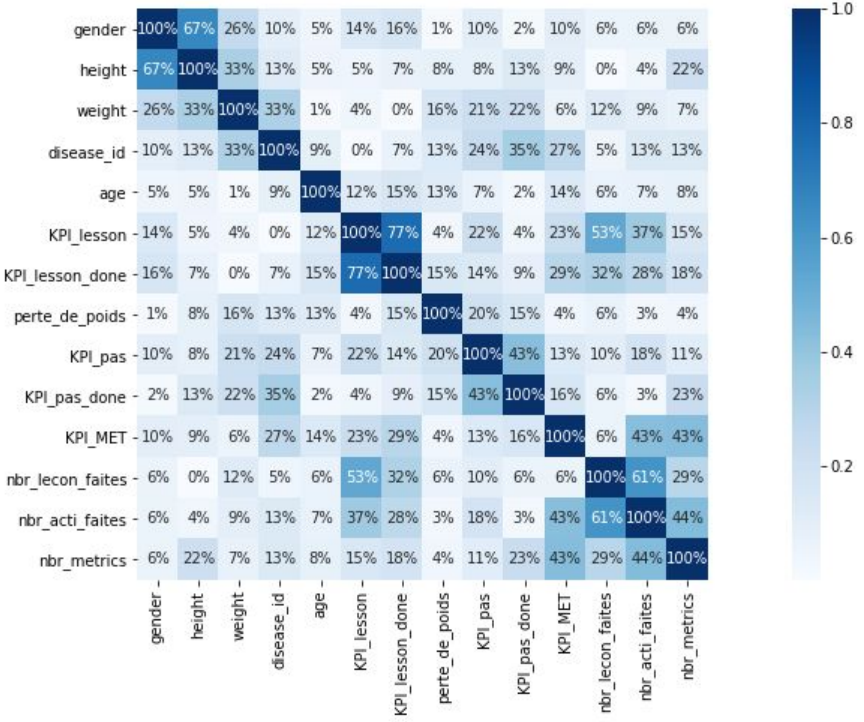
III. Matrice de corrélation



Chaque pourcentage correspond au “lien statistique” entre les deux séries de données (i.e le rapport de la covariance sur le produit des écarts-types)

$$r = \left| \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y} \right|$$

III. Matrice de corrélation



KPI_lesson : Pourcentage de leçons effectuées

KPI_lesson_done : 1 si KPI_lesson > 0,66 - 0 sinon

perte_de_poids : pourcentage de perte de poids entre la première valeur rentrée et la dernière

KPI_pas : nombre de pas effectués sur les 120 premiers jours (si moins de 120 jours dans le programme, extrapolation avec la moyenne)

KPI_pas_done : 1 si KPI_pas >840 k - 0 sinon

KPI_MET : nombre de MET effectués sur les 120 premiers jours (si moins de 120j dans le programme, extrapolation avec la moyenne)

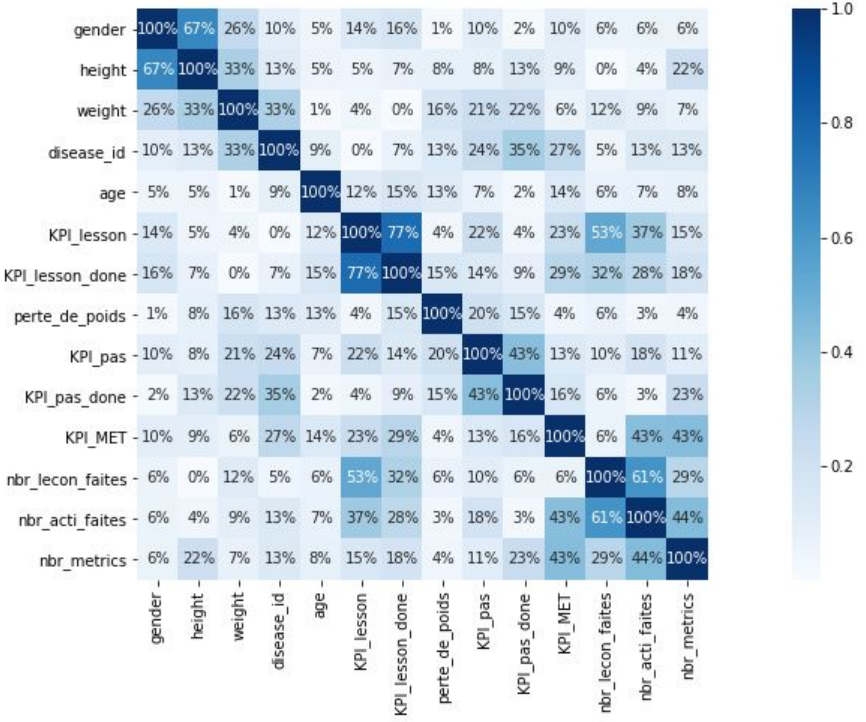
KPI_MET_done : 1 si KPI_MET > 9000 - 0 sinon

nbr_lesson_faites : Nombre de leçons faites par les utilisateurs sur les 3 premières semaines

nb_actis_faites : Nombre d'activités faites par user sur les 3 premières semaines

nbr_metrics : Nombre d'entrées dans le fichier metrics par utilisateur sur les 3 premières semaines

III. Matrice de corrélation



Corrélations importantes (>30%) entre :

Disease_id/KPI_pas_done :
Le nombre de pas réalisé dépend de manière globale de la cohorte étudiée.

KPI_lesson/nbr_lesson_faites :
La réussite du KPI des leçons est corrélée à l'activité sur les 3 premières semaines en terme de leçons.

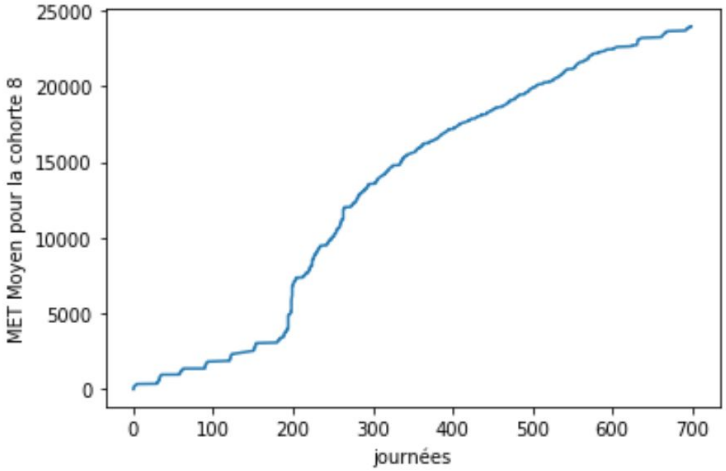
KPI_MET/nbr_metrics et nbr_actis_faites :
La réussite du KPI des MET est corrélée à l'activité sur les 3 premières semaines.

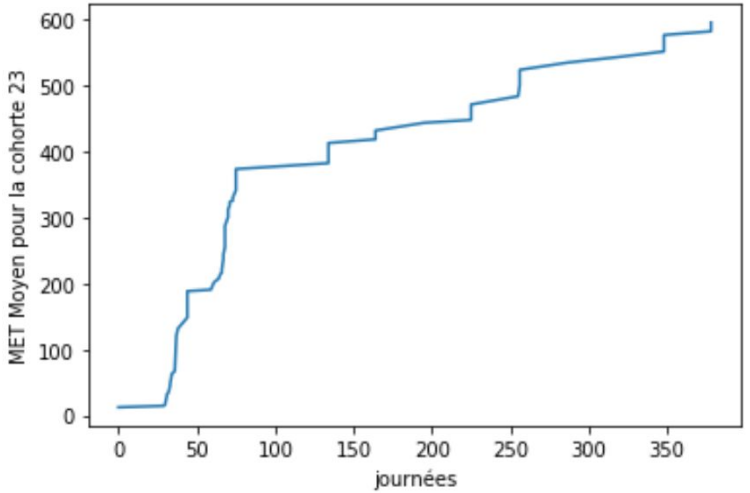
nbr_lesson_faites/nbr_metrics/nbr_actis_faites :
L'implication dans les 3 premières semaines est assez global dans tous les aspects du programme.

III. KPI moyen par cohorte

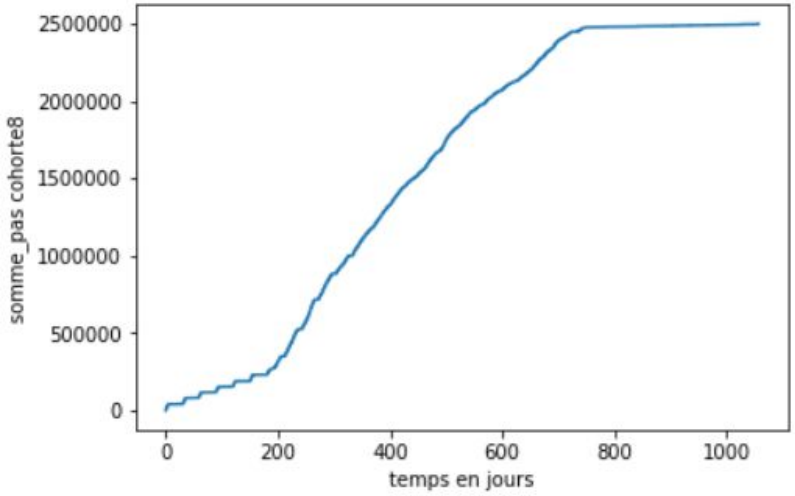
Index	gender	height	weight	age	KPI_lesson	perte_de_poids	KPI_pas	KPI_MET	user_nb
1.0	1.5	173.16	80.548	36.28	0.441939	2.07856	827090	3676.29	28
2.0	1.22222	166.222	74.5556	41.875	0.419643	16.5143	3.07935e+06	1386.5	9
3.0	1.22222	165.778	67.4444	46.2857	0.32963	1.43665	1.83669e+06	2750.68	9
5.0	1.02941	164.765	59.5	47.1379	0.315476	1.64264	4.43673e+06	621.136	34
8.0	1	163.846	73.9692	49.375	0.531136	1.55746	1.06598e+06	2371.24	13
10.0	1.42857	169	70.7	37.4286	0.35	6.04532	1.25499e+06	3537.6	7
11.0	1	163.375	74.375	53.75	0.489796	3.37725	877707	1834.25	8
12.0	1.5	167.75	72.5	47	0.440476	1.42857	631355	1420.36	4
14.0	2	191	112.2	45	0.2	2.68864	1.22597e+06	nan	1
18.0	1.625	172.875	72.2875	49.25	0.5	-1.08568	960275	1298.83	8
19.0	1.41667	168.333	75.7167	43.9091	0.34127	2.6153	719054	463.917	12
20.0	1	166.2	122.32	41.2	0.198718	-0.0271772	788212	1156.87	5
21.0	1	166.667	84.7167	48.3333	0.646199	-6.60645	568838	655.2	6
22.0	1.42857	158.286	60.35	21.5	0.444444	nan	1.47826e+06	1225.73	7
23.0	1.3	166.9	120.11	34.1111	0.246154	-0.511767	432051	453.643	10
24.0	1.25	165.083	77.5	44.75	0.305861	0.938912	nan	1151.13	12
25.0	1.53846	163.75	85.7769	53.3333	0.147518	1.40717	1.36176e+06	2060	13

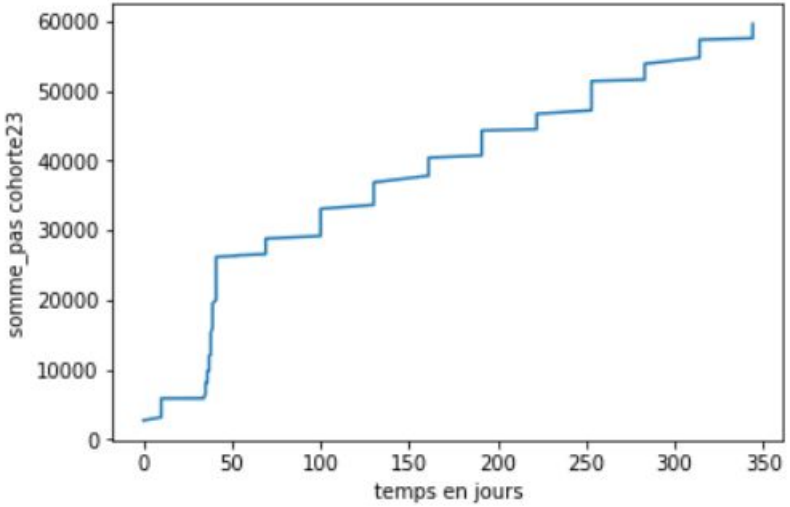
III. Temporalité & KPI par cohorte





III. Temporalité & KPI par cohorte



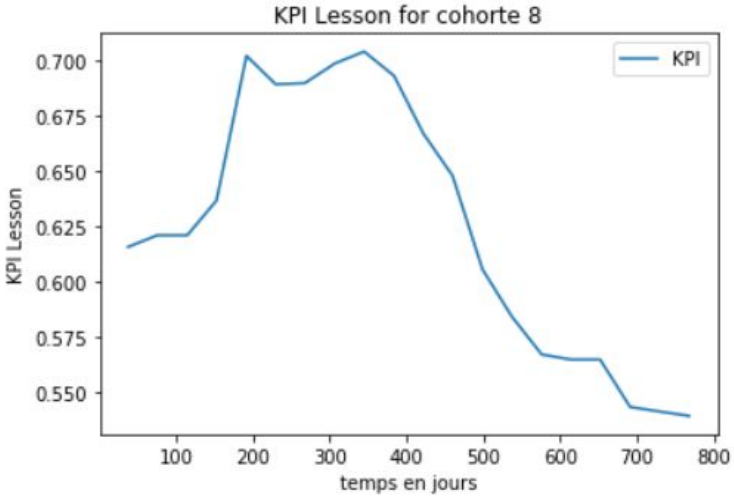


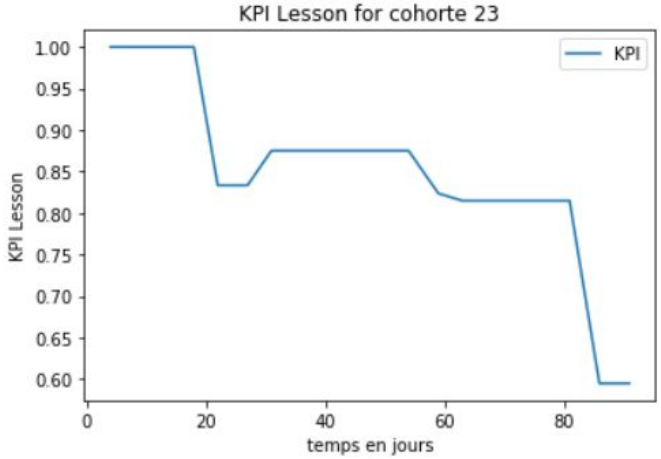
Dynamique similaire des deux graphiques sur la même échelle de temps

Cohorte 8 : 13 users

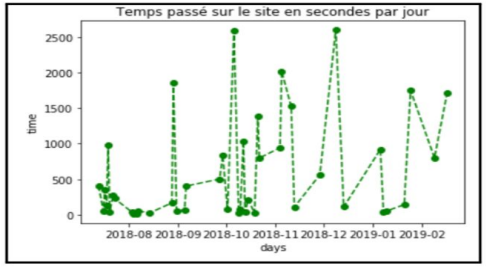
Cohorte 23 : 10 users

III. Temporalité & KPI par cohorte





III. Résultat traitement JSON



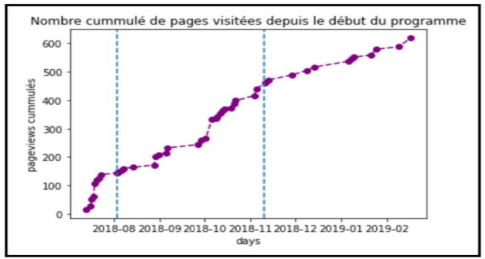


FIGURE 4 – tableaux récapitulatifs pour le user 588

Caractère aléatoire de la connection -> corrélation avec les discussion avec le coach ?

Période d'activité plus intense entre les deux barres bleues qui correspondent à 21j et 120j.

Traitement sur 3 utilisateurs mais applicable à n'importe quel fichier.

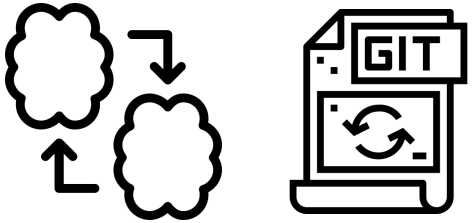
III. Synthèse des résultats

> **Matrice de corrélation** : implication sur 3 semaines importante.

> **Moyennes par cohorte et étude de la temporalité** : phénomènes de groupe.

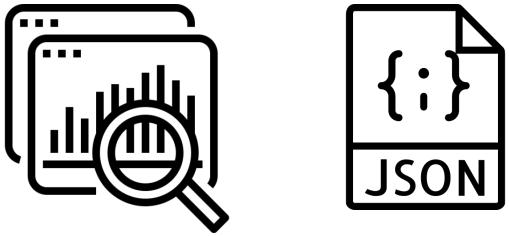
> **Traitement .JSON** : période d'activité plus intense entre les trois premières semaines et les trois premiers mois.

IV. Suite donnée au projet



Passation de savoir grâce au Git :
https://gitlab.com/latitudes-exploring-tech-for-good/stimul/1819_stimul/

- Programmes sans les données (médicales)
- Rapport .pdf explicatif de nos résultats

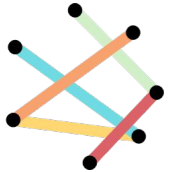


Analyse des données par Stimul :

- Confirmation ou non des intuitions par analyse des résultats
- Reprise des programmes pour l'étendre à d'autres set de données (json)
- Extension du projet sous forme de dashboard

Conclusion

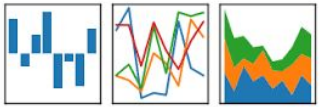
- Découverte de l'univers Tech For Good grâce à Latitudes



latitudes
exploring tech
for good_

- Compétences en analyse de données via Pandas

pandas
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

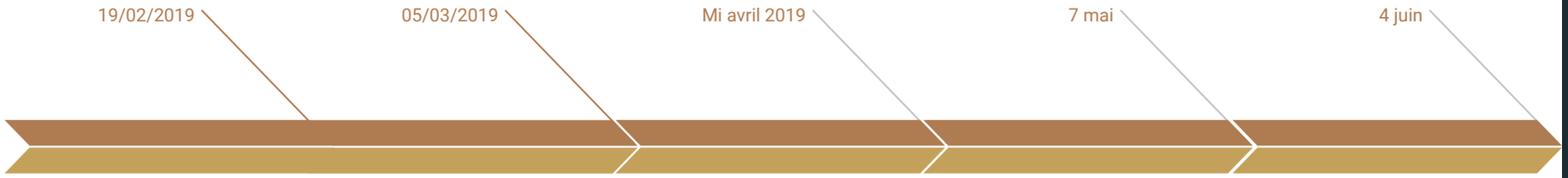


- Gestion de projet de code en groupe

Merci de votre attention !
Des remarques ?

https://gitlab.com/latitudes-exploring-tech-for-good/stimul/1819_stimul/

IV. Retour sur le planning depuis le début de l'année



Exploration

Collecte des données, rencontre des acteurs

Montée en compétences et nettoyage des données

Exploration des données

Prototypage et test des hypothèses

Analyse des données et corrélation

Mise en forme du livrable

Communication des résultats

Finitions

Finition des documents et préparation de la soutenance finale.