IRONHACK FINAL PROJECT

CIVIC TECH WHO'S WHO

2023 DAFT0410 - TATIANA DE FERAUDY

TODAY'S PRESENTATION

- The 'business' case
- Project objectives, planning and tasks
- Data collection & structure
- Data cleaning
- ERD
- Insights
- Modeling
- Takeaways and next steps

‘BUSINESS’ CASE

Knight Foundation: Trends in Civic Tech

THEMES
CLUSTERS
ORGANIZATIONS

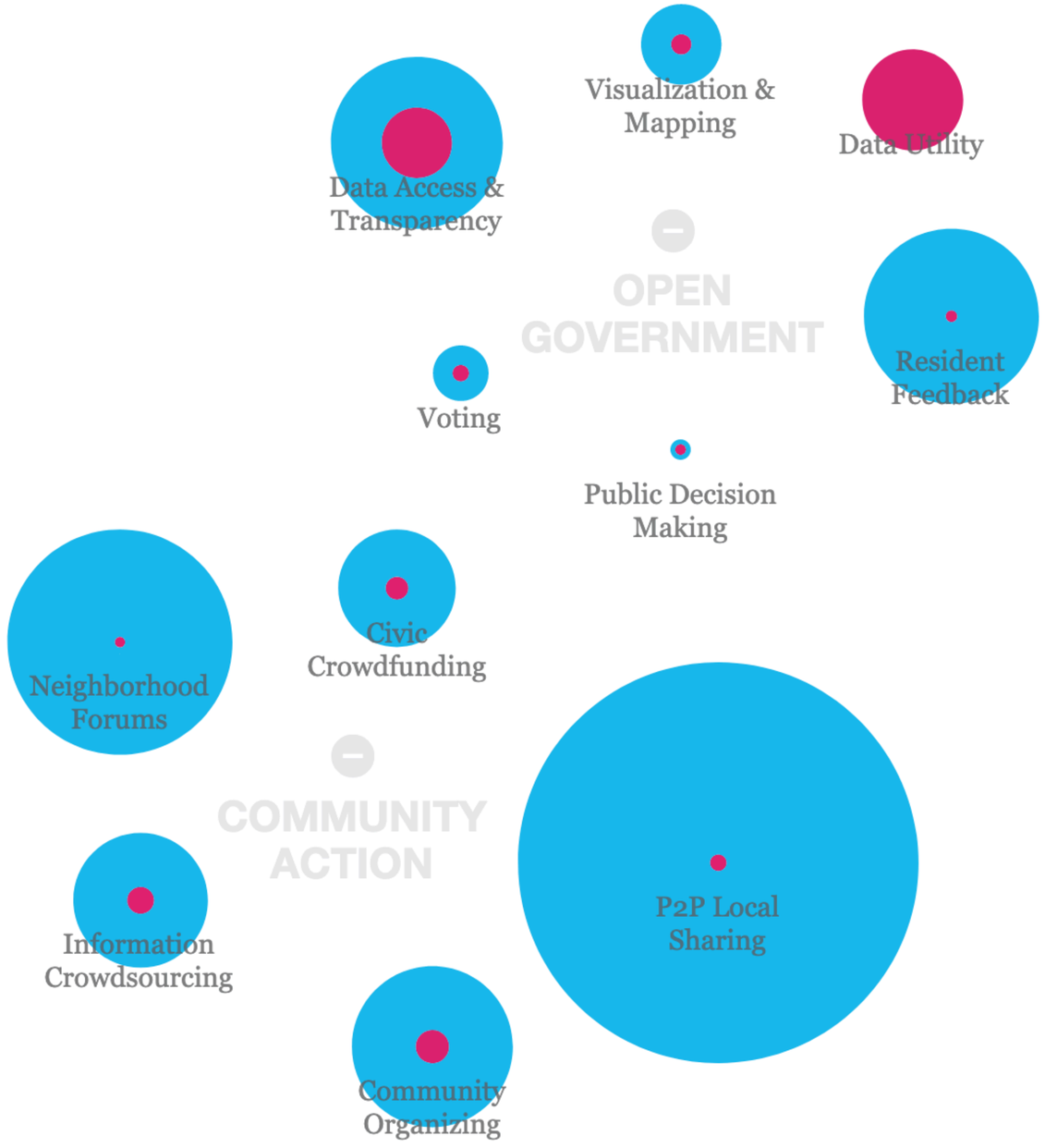
Learn more about the study

Get the data (xls)

Anything missing? Send us feedback

Tweet

Find organizations



READING THE CHART

Investment Type

- Private
- Grant
- No investment

Investment Size ($)
Jan 2011 — Dec 2013

Small — Large

Network analysis by Quid
Visualization by Fathom

Open
Government
Partnership

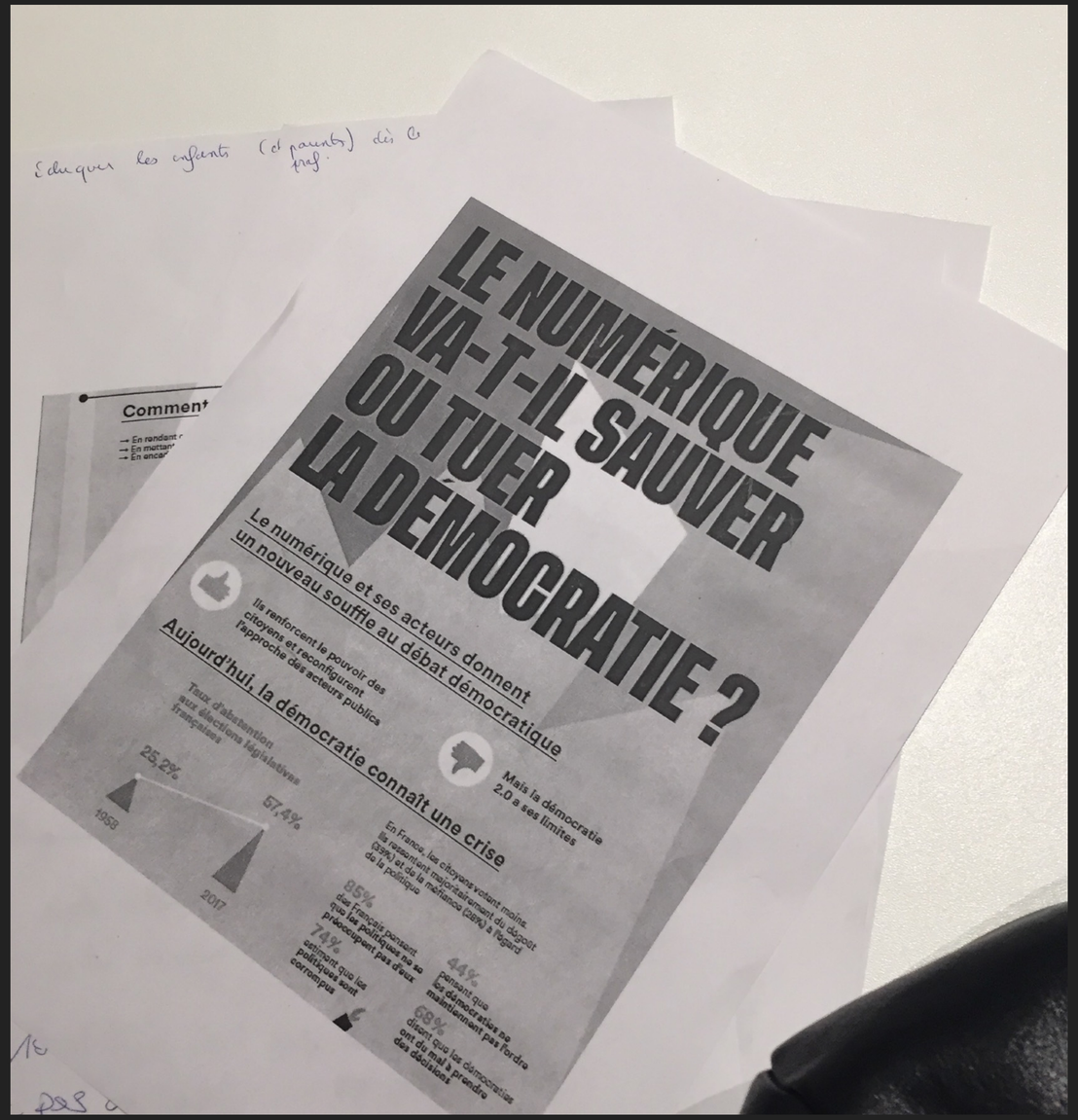
PROJECT

A PHD PROJECT

- Who built the civic tech public problem and how ?
- Qualitative methods and collection of 'grey' literature

AN IRONHACK PROJECT

- Quantitative/ 'big data' approach
- A replicable framework for data collection and analysis



PROJECT PLANNING

Final_project_Kanban | Workspace visible | Board | Power-Ups | Automation | Filter | TF | Share

Data collection

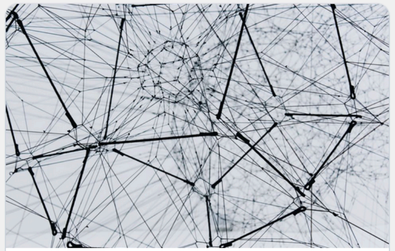
Online data

Combine with PhD data

+ Add a card

Data cleaning

Database construction

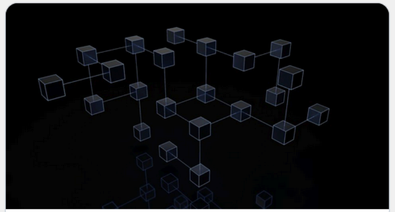


Exploratory data analysis

+ Add a card

Analysis & Presentation

Data visualisation



Data analysis and modeling

+ Add a card

Stuff to remember

Personal data - pseudonymize

+ Add a card

Bonuses

Important : improve scraper to get previous experience

improve import code with enumerate in columns

+ Add a card

Presentation 1 / 3

Slides and github repo

1

+ Add a card

Done

Done

Done

1

Make the linkedin scraper work to produce json files of profiles

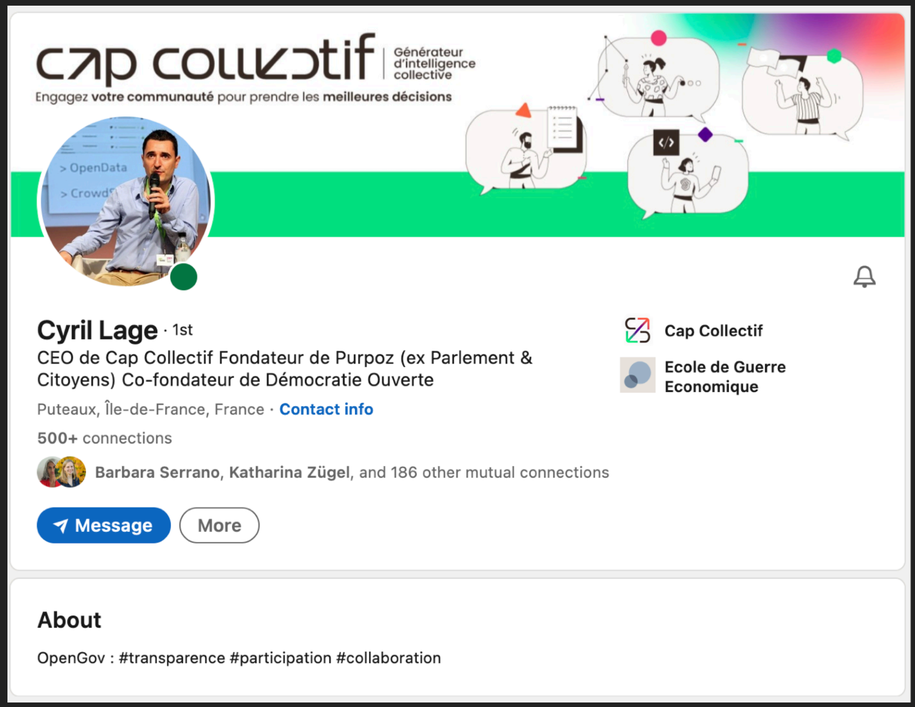
Write code to make a dataframe from json files scraped from linkedin

+ Add a card

31 MARCH 2023

DATA

DATA COLLECTION



Cyril Lage · 1st
CEO de Cap Collectif Fondateur de Purpoz (ex Parlement & Citoyens) Co-fondateur de Démocratie Ouverte
Puteaux, Île-de-France, France · **Contact info**
500+ connections
Barbara Serrano, **Katharina Zügel**, and 186 other mutual connections

Message | More

Cap Collectif
Ecole de Guerre Economique

About

OpenGov : #transparence #participation #collaboration

Experience

CEO
Cap Collectif
Jul 2014 - Present · 9 yrs
Paris Area, France

Startup experte dans le domaine de l'intelligence collective qui propose une plateforme de consultation en ligne aux organisations publiques et privées.

Président
Parlement & Citoyens
Feb 2013 - Present · 10 yrs 5 mos
Paris

Co-Fondateur
Démocratie Ouverte
Sep 2011 - Present · 11 yrs 10 mos
France / Québec / Tunisie / Suisse / Belgique

Collectif francophone dédié à la promotion de la démocratie ouverte (open government) - #transparence #participation #collaboration

Associé
Spin Partners
Nov 2002 - Jun 2012 · 9 yrs 8 mos
Paris Area, France

Responsable du développement et des partenariats

Education

Ecole de Guerre Economique
2000 - 2001

IEP Toulouse
DESS
1999 - 2000

UNIVERSITE D'AUVERGNE
DEUG, Licence et Maîtrise de droit privé
1996 - 1999

Projects

Membre de démocratie ouverte
Jan 2012 - Present

Show project

Démocratie ouverte est un collectif de citoyens issus de plusieurs pays francophones. Toutes passionnées par le service public et le numérique, ces personnes sont convaincues que le gouvernement ouvert est une s… ...see more

Other contributors

8

DATA COLLECTION


```
[1]
1  import pathlib
2  import pandas as pd
3  from bs4 import BeautifulSoup
4  import pickle
5  import json
6  import requests
7  import csv
8  import os
9  from linkedin_api import Linkedin
```

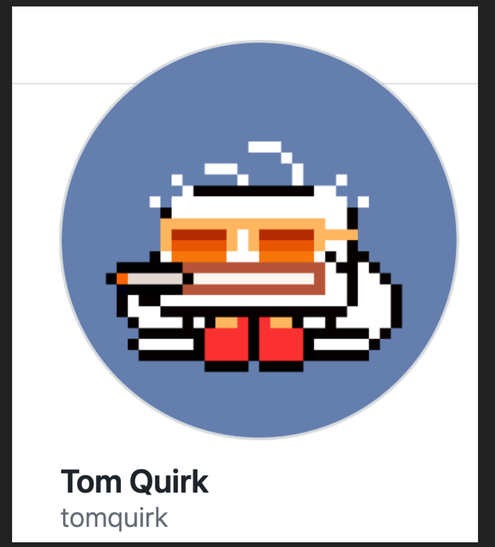
```
[2]
1  a="https://www.linkedin.com/in/brachetantoine/  https://www.linkedin.com/in/maxbarbier/
2  list_urls=a.split()
3  list_urls
```

Outputs are collapsed ···

```
[3]
1  for i in list_urls:
2      id= i.split('/in/')[1]
3      id= id.split('/')[0]
4      print(id)
```

Outputs are collapsed ···

Initializing and splitting URLs



DATA COLLECTION


```python
class Adresse:
    def __init__(self, nom, url, suffixe=None):
        self.nom = nom
        self.url = url
        self.suffixe = suffixe
```

✓ 0.0s

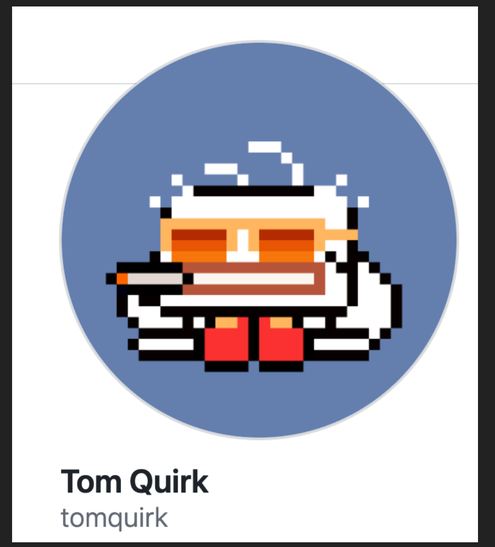
```python
password= getpass.getpass()
api = Linkedin("tatianadeferaudy@yahoo.fr", password)
```

✓ 3.9s

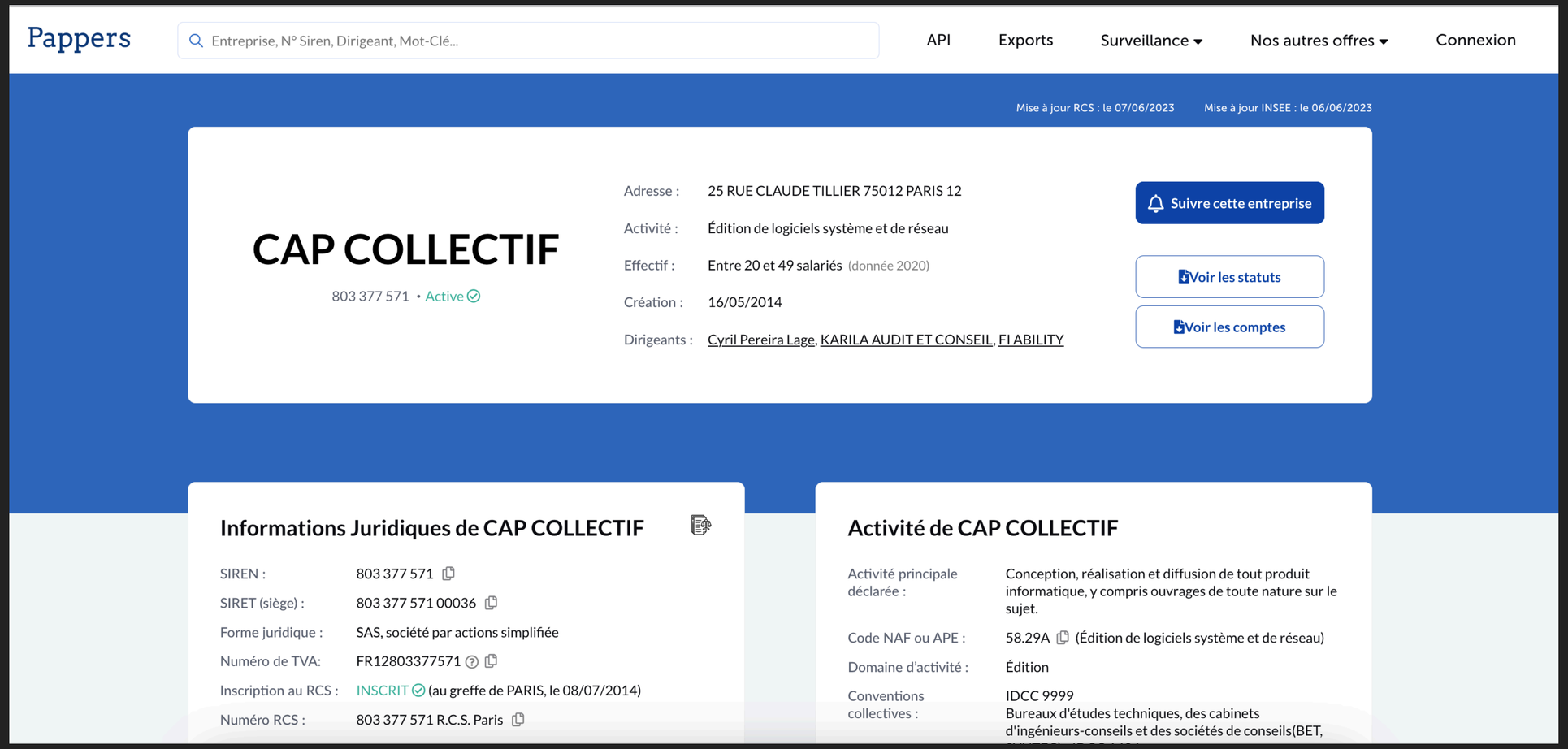
```python
for i in list_urls:
    id=i.split('/in/')[1]
    suffixe= id.split('/')[0]
    profile_content = api.get_profile(suffixe)
    #print(profile)
    # Sauvegarder le profil dans un fichier avec un nom différent
    filename = suffixe + '.json'
    filepath = os.path.join('//Users/tatianadeferaudy/Desktop/Bacasable/IronHack/final_project',
                            filename)
    print(filepath)
    with open(filepath, 'w', encoding='utf-8') as f:
        json.dump(profile_content, f)
```

//Users/tatianadeferaudy/Desktop/Bacasable/IronHack/final_project/brachetantoine.json
//Users/tatianadeferaudy/Desktop/Bacasable/IronHack/final_project/maxbarbier.json
//Users/tatianadeferaudy/Desktop/Bacasable/IronHack/final_project/frankescoubes.json
//Users/tatianadeferaudy/Desktop/Bacasable/IronHack/final_project/martinduval.json
//Users/tatianadeferaudy/Desktop/Bacasable/IronHack/final_project/juliealbet.json

Getting the profile information : api.get_profile()



DATA COLLECTION



DATA COLLECTION/ STRUCTURE


```
1  data_companies['citility'][1]
2  # [1] provides company information : SIREN, juridical form,
3  # TVA number, inscription RCA, n°, social capital
4  # it's a dataframe
```

	0	1
0	SIREN :	802 503 276
1	SIRET (siège) :	802 503 276 00023
2	Forme juridique :	SAS, société par actions simplifiée
3	Numéro de TVA:	FR07802503276
4	Inscription au RCS :	INSCRIT (au greffe de LYON, le 12/05/2014)
5	Numéro RCS :	802 503 276 R.C.S. Lyon
6	Capital social :	39 620,00 €

Nested dataframes in our dictionary

DATA COLLECTION/ STRUCTURE


```
new_df=pd.concat([pd.DataFrame(data_companies['citility'][0][0]),
                  pd.DataFrame(data_companies['citility'][1][0]),
                  pd.DataFrame(data_companies['citility'][2][0][0:3])],
                  axis=0).reset_index(drop=True)
new_df.columns=['ind']
for i in list(data_companies.keys()):
    a= pd.concat([pd.DataFrame(data_companies[str(i)][0]),
                  pd.DataFrame(data_companies[str(i)][1]),
                  pd.DataFrame(data_companies[str(i)][2][0:3])],
                  axis=0).reset_index(drop=True)
    a.columns=['ind', i]
    new_df= new_df.merge(a, how='outer')
```

Recreating the structure we need

DATA COLLECTION/ STRUCTURE

cyril-lage-45a4967.json — jsons (git: main)


```
{
    "summary": "OpenGov : #transparence #participation #collaboration",
    "industryName": "Public Policy Offices",
    "lastName": "Lage",
    "locationName": "France",
    "student": false,
    "geoCountryName": "France",
    "geoCountryUrn": "urn:li:fs_geo:105015875",
    "geoLocationBackfilled": false,
    "elt": false,
    "birthDate": {
        "month": 10,
        "year": 1976,
        "day": 23
    },
    "industryUrn": "urn:li:fs_industry:79",
    "firstName": "Cyril",
    "entityUrn": "urn:li:fs_profile:ACoAAAFRJAYBLIpu6yVKFiYIRZijrUKSuOUb8JQ",
    "geoLocation": {
        "geoUrn": "urn:li:fs_geo:103424094"
    },
    "geoLocationName": "Puteaux, Île-de-France",
    "location": {
        "basicLocation": {
            "countryCode": "fr"
        }
    },
    "headline": "CEO de Cap Collectif\nFondateur de Purpoz (ex Parlement & Citoyens)\nCo-fondateur de
Démocratie Ouverte",
    "displayPictureUrl":
"https://media.licdn.com/dms/image/C5603AQGjUuPvsKiQqQ/profile-displayphoto-shrink_",
    "img_100_100":
```


From a json file with nested dictionaries

DATA COLLECTION/ STRUCTURE


```
def create_profile(x):
    with open('../data/jsons/'+str(x)) as f:
        dict1 = json.load(f)
    list_col= ['experience', 'education', 'languages']
    for n in list_col:
        if n in dict1:
            for i in range(len(dict1[n])):
                dict1[str(n+str(i+1))]= dict1[n][i]
    data = pd.DataFrame.from_dict(dict1, orient='index').T
    return data
```

Creating and concatenating dataframes built from the json file

CLEANING

DATA CLEANING/ BASIC

[138]
```
finance_df=pd.DataFrame(data_companies['citility'][3])
for i in list(data_companies.keys()):
    if len(data_companies[str(i)][3]) > 10:
        b= pd.DataFrame(data_companies[str(i)][3]).set_index("Performance").add_suffix('_'+i)
        finance_df=finance_df.merge(b, how='outer', on="Performance")
finance_df.drop(columns=['2017', '2016'], inplace=True)
```

[139]
```
finance_df.head()
```

	Performance	2017_citility	2016_citility	2019_voxcracy	2018_voxcracy	2017_voxcracy	2016_voxcracy	2020_LLL_2	2
0	Chiffre d'affaires (€)	NaN	30,7K	46,3K	16,7K	1,74K	0	473K	
1	Marge brute (€)	NaN	527K	46,3K	157K	75,9K	NaN	473K	
2	EBITDA - EBE (€)	NaN	-295K	-31,5K	-123K	-24,4K	-1,08K	24,7K	
3	Résultat d'exploitation (€)	NaN	-296K	-38,5K	-130K	-30,3K	-4,91K	-42,7K	
4	Résultat net (€)	-562K	-238K	-39K	-112K	-23,4K	-4,91K	-44,9K	

5 rows × 75 columns

Numbers stored as text

DATA CLEANING/ BASIC


```python
import re

def fix_columns(x):
    x=str(x)
    if x== 'nan':
        return 0
    elif 'K' in x:
        if ',' in x:
            return int(re.split('[,K]', x)[0]+re.split('[,K]', x)[1]+'0'*(3-len(re.split('[,K]', x)[1])))
        else:
            return int(x.replace('K', '000'))
    elif 'M' in x:
        if ',' in x:
            return int(re.split('[,M]', x)[0]+re.split('[,M]', x)[1]+'0'*(6-len(re.split('[,M]', x)[1])))
        else:
            return int(x.replace('M', '000'))
    else:
        return x
```

Using regex to clean that column

DATA CLEANING/ BASIC


```
1  finance_df2.head(10)
```

index	chiffre_daffaires_e	marge_brute_e	ebitda___ebe_e	resultat_dexploitation_e	resultat_net_e
2017_citility	0	0	0	0	-562000
2016_citility	30700	527000	-295000	-296000	-238000
2019_voxcracy	46300	46300	-31500	-38500	-39000
2018_voxcracy	16700	157000	-123000	-130000	-112000
2017_voxcracy	1740	75900	-24400	-30300	-23400
2016_voxcracy	0	0	-1080	-4910	-4910
2020_LLL_2	473000	473000	24700	-42700	-44900
2019_LLL_2	462000	462000	160000	95200	62400
2018_LLL_2	300000	300000	89700	25900	19800
2017_LLL_2	0.0	0	0	0	0

10 rows × 47 columns

And here we have nice numbers to apply calculations on !

DATA CLEANING/ CREATING CATEGORIES


```python
def clean_inscriptions(x):
    x=str(x)
    if "INSCRIT" in x:
        return 1
    else:
        return 0
```

```python
associations["inscription_rna"]=associations['inscription_au_rna'].apply(clean_inscriptions)
```

```python
def get_date(x):
    x=str(x)
    pattern=r"\d{1,5}/\d{2,5}/\d{2,5}"
    a= re.findall(pattern, x)
    a= ''.join(a).strip()
    return a
```

```python
associations["date_inscr"]=pd.to_datetime(associations['inscription_au_rna'].apply(get_date), dayfirst=True)
```

Creating binary categories, getting dates, changing data types

DATA CLEANING/ CREATING CATEGORIES


```python
# recoding location columns
# note : this type of formatting flattens elements with different locations
# giving priority to the french one

def recoding_location(x):
    x=str(x)
    if ("Paris" in x) or ("PAris" in x) or ("Montreuil" in x) or ("Puteaux" in x):
        return 'Paris Metropolitan Region'
    elif ("Brussels" in x) or ("Bruxelles" in x):
        return 'Brussels Metropolitan Region'
    elif "Berlin" in x:
        return 'Berlin Metropolitan Region'
    elif "Nantes" in x:
        return 'Nantes Metropolitan Region'
    elif "Bordeaux" in x:
        return 'Bordeaux Metropolitan Region'
    elif "Lyon" in x:
        return 'Lyon Metropolitan Region'
    elif "Marseille" in x:
        return 'Marseille Metropolitan Region'
    elif "Lille" in x:
        return 'Lille Metropolitan Region'
    else:
        return x

# possible improvement with geopy library

# replace " France" (if it is only the word, not Ile de France) by nothing?
```

Recoding (1) : geographical location - useful for Tableau

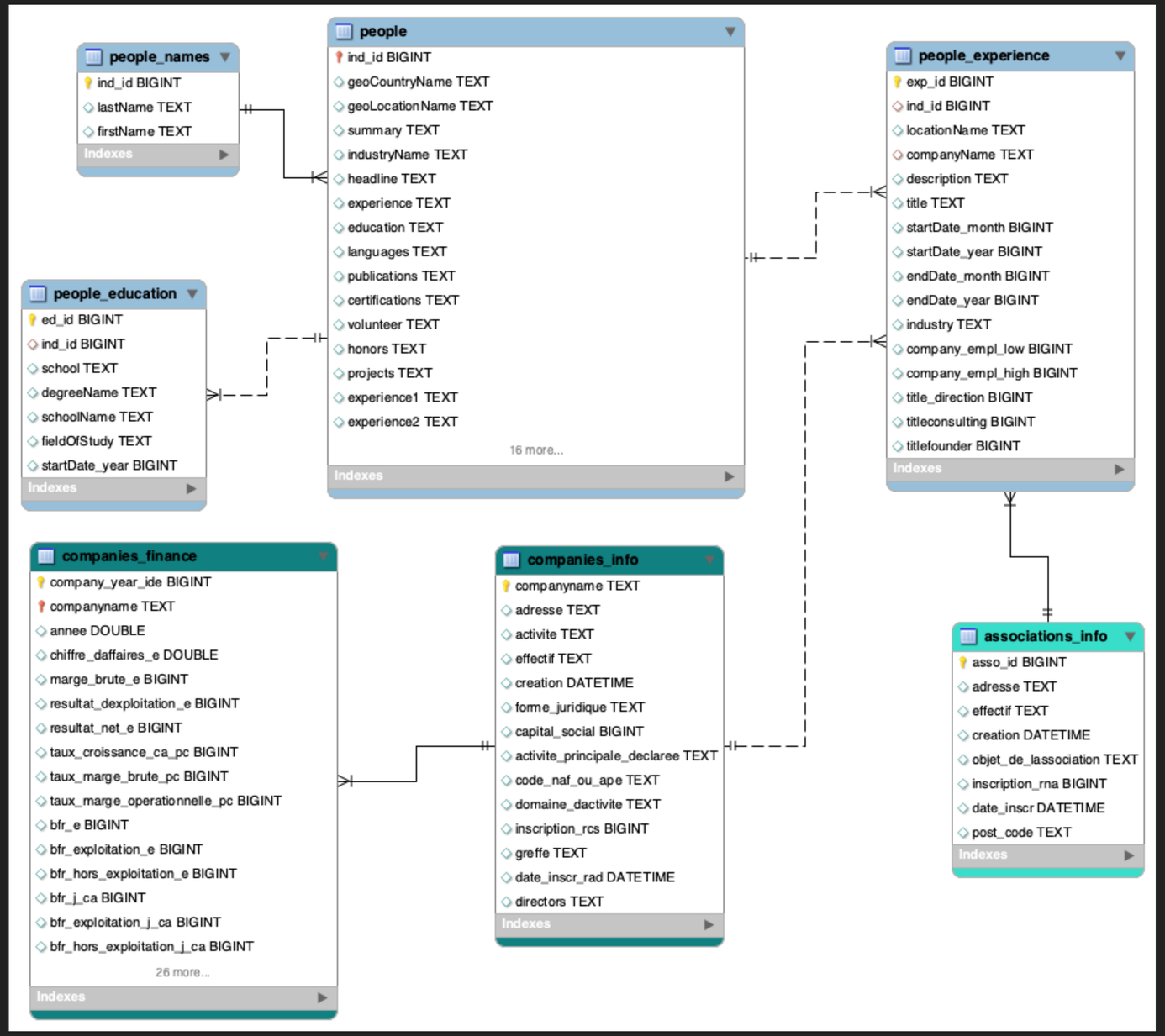
DATA CLEANING/ CREATING CATEGORIES


```
def recoding_title_dir(x):
    x=str(x).lower()
    dir=["ceo", "coo", "cfo", "président", "directeur", "directrice", "director",
        "cto", "cpo", "general manager", "president", "head of"]
    if any([y in x for y in dir]):
        return 1
    else:
        return 0

def recoding_title_cs(x):
    x=str(x).lower()
    cs= ["consultant", "conseiller", "conseillère"]
    if any([y in x for y in cs]):
        return 1
    else:
        return 0

def recoding_title_fond(x):
    x=str(x).lower()
    fond=["founder", "fondateur", "fondatrice"]
    if any([y in x for y in fond]):
        return 1
    else:
        return 0
```

Recoding (2) : understanding titles/ management positions

ERD

DATABASE ENTITY RELATIONSHIP DIAGRAM



46 individuals at first

151 for modeling

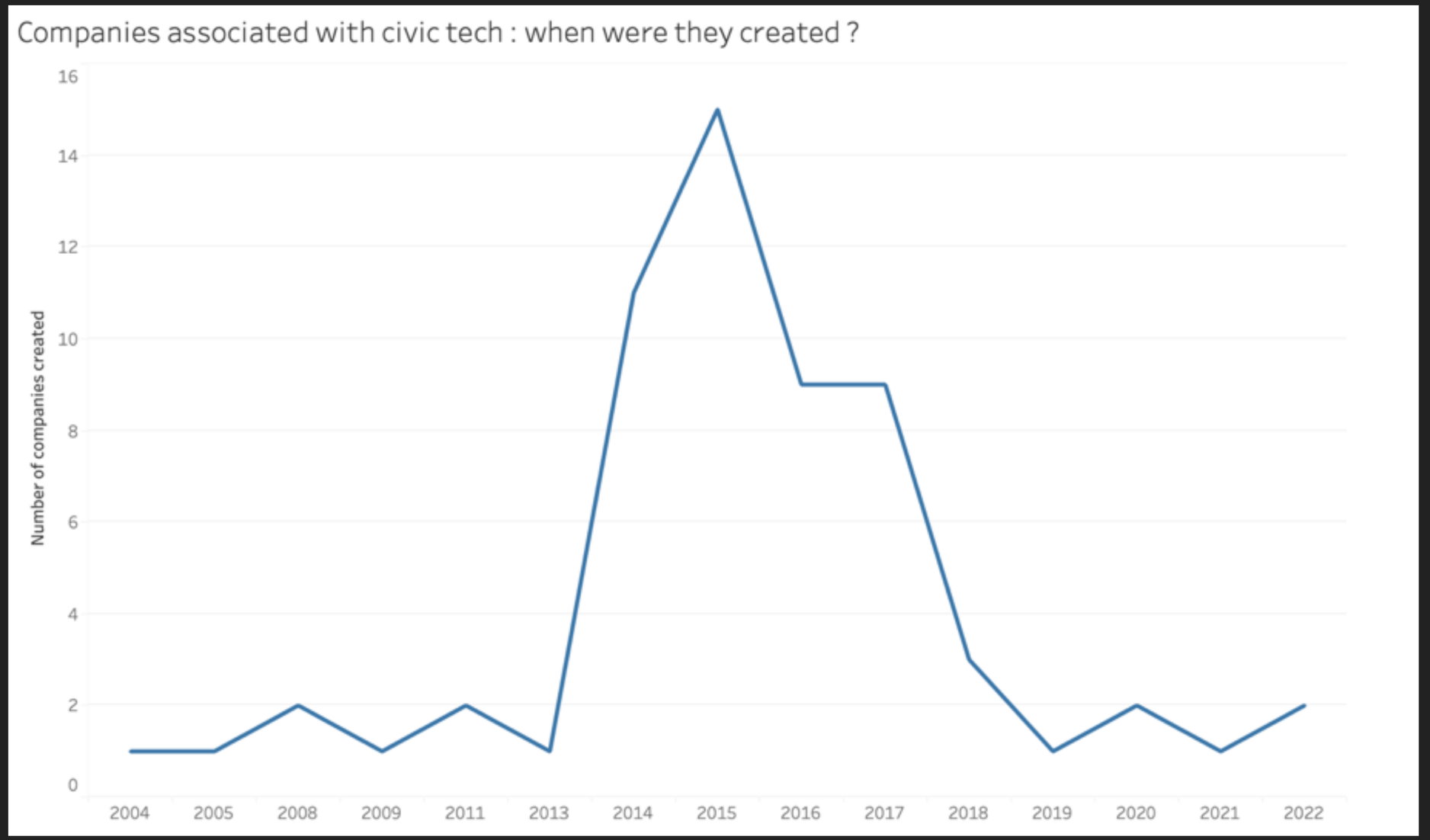
~ 300 to add

76 companies at first

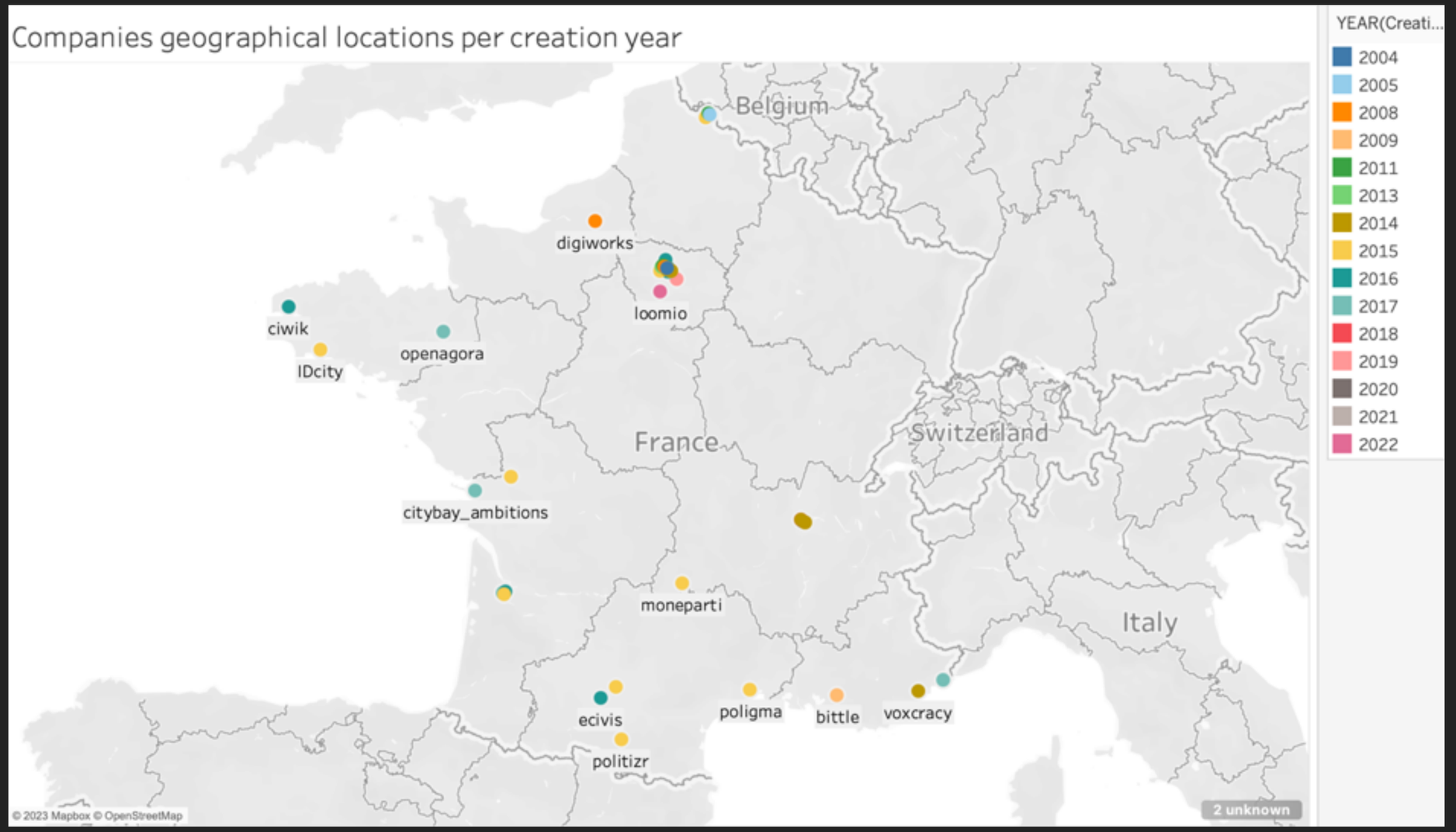
~ 30 to 50 to add

INSIGHTS

DATA ANALYSIS & VISUALIZATION : TABLEAU

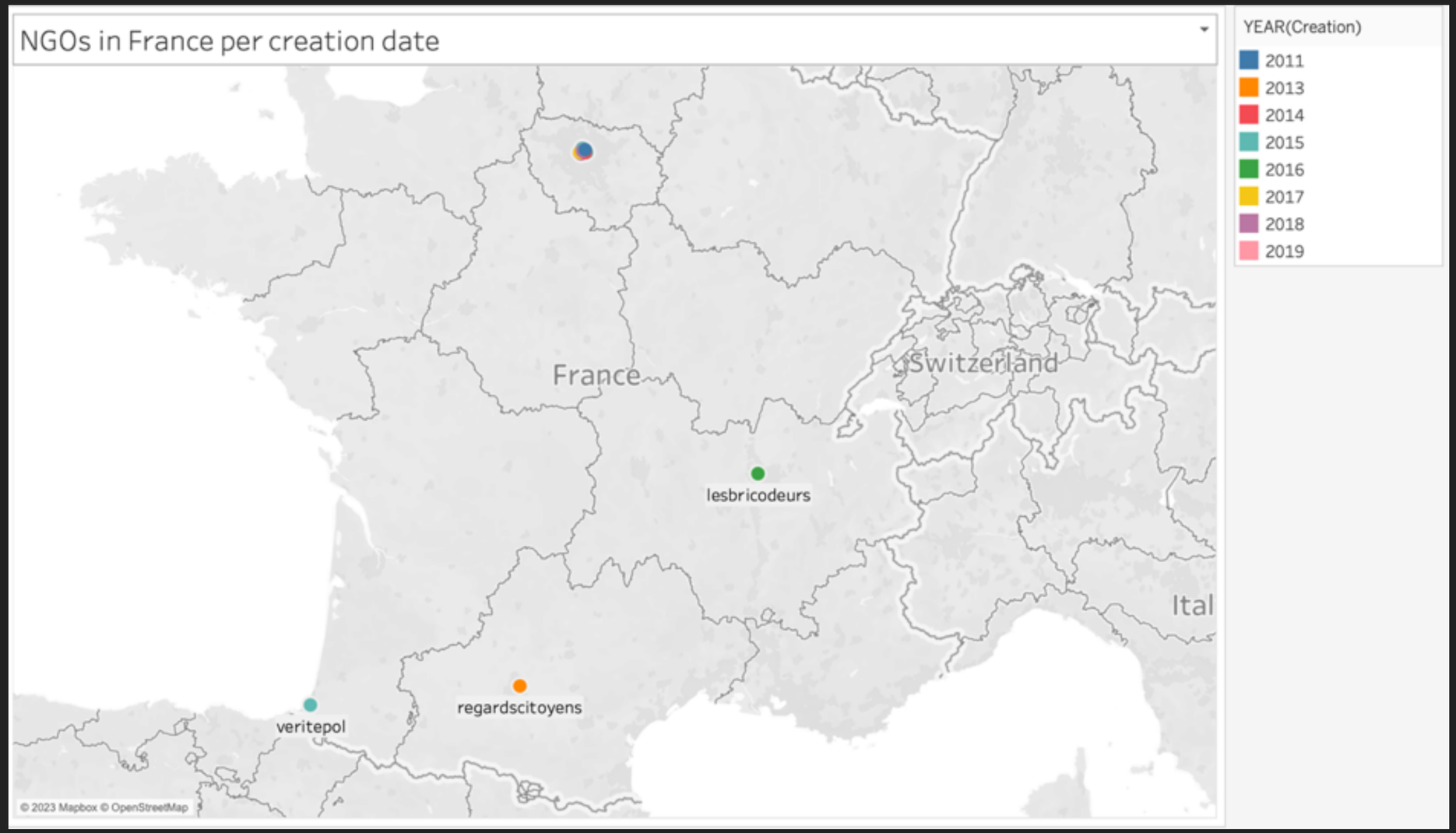


DATA ANALYSIS & VISUALIZATION : TABLEAU



Companies are concentrated in Paris - but there are « old » companies outside of Paris

DATA ANALYSIS & VISUALIZATION : TABLEAU

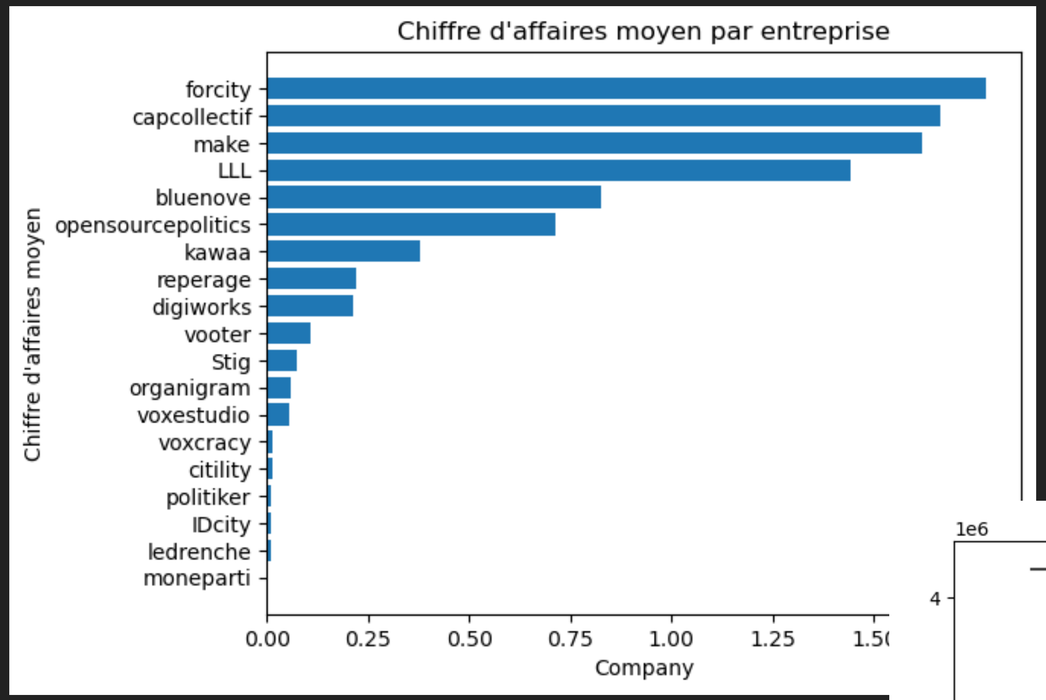


NGOs also concentrated in Paris

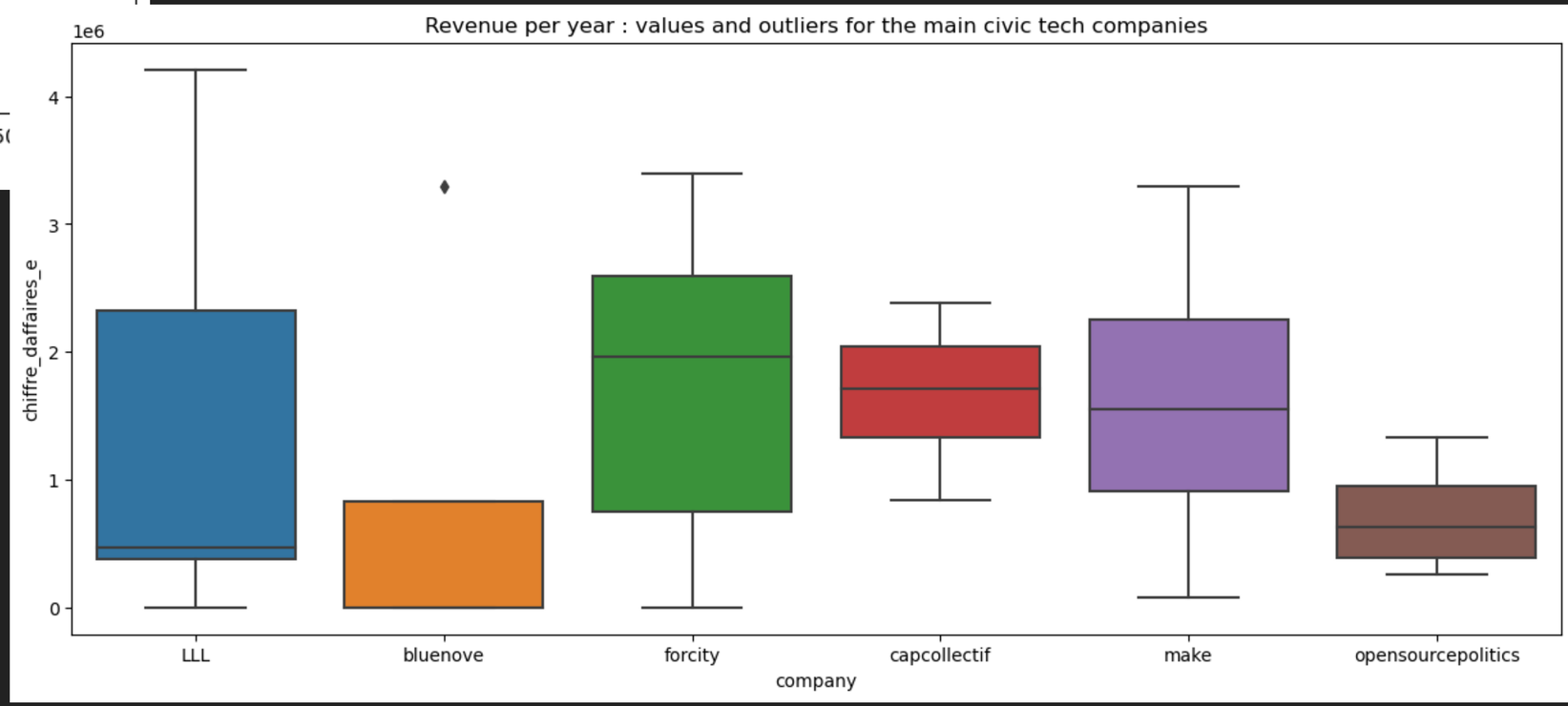
DATA ANALYSIS & VISUALIZATION : TABLEAU

Bankrupcy related to square meter price ?

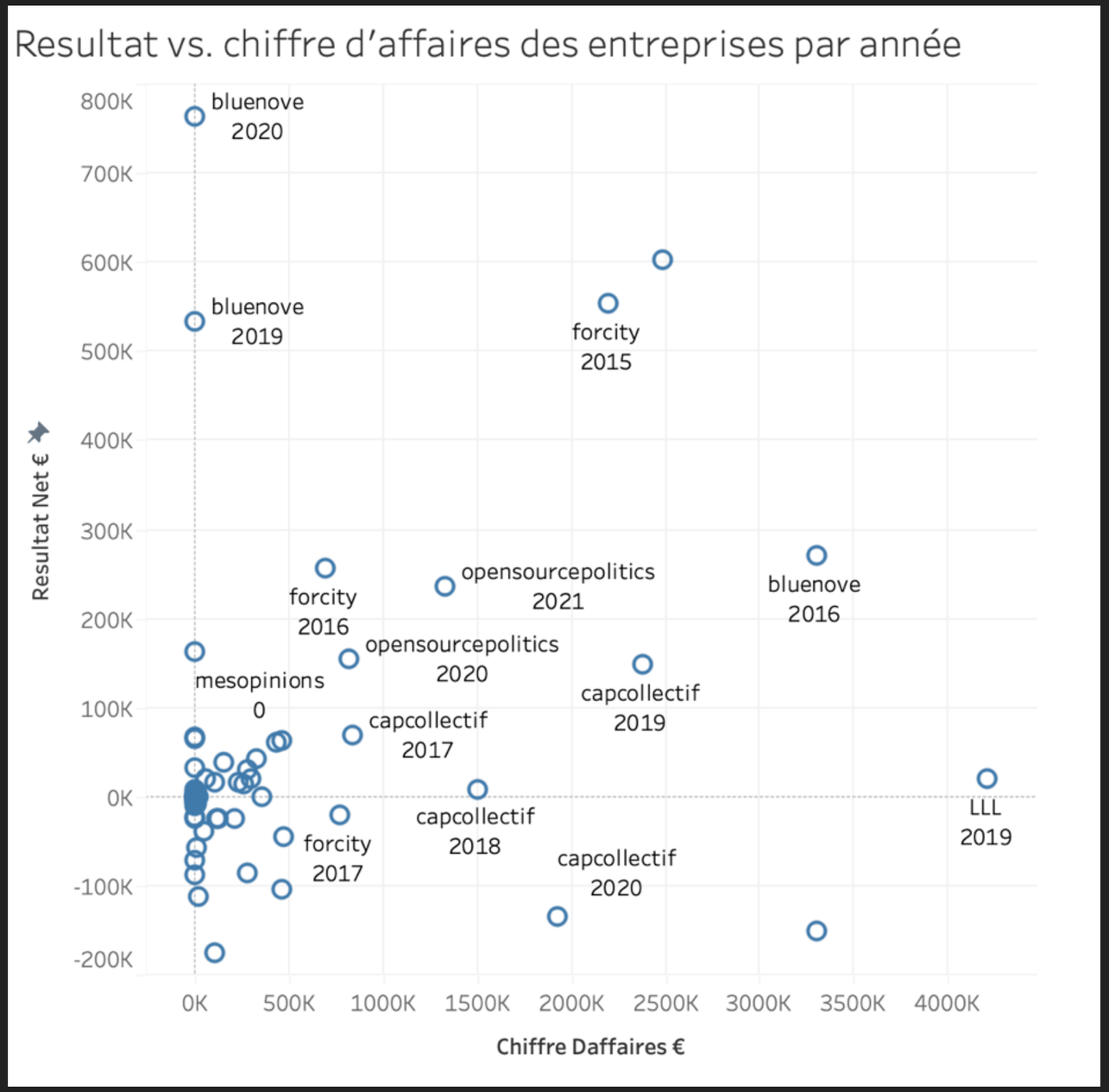
DATA ANALYSIS & VISUALIZATION: PYTHON 2



Identifying interesting variables to work with and their limits



DATA ANALYSIS & VISUALIZATION: TABLEAU 1



There are 50 ways to show the same thing…

but you may get some additional insight along the way

DATA ANALYSIS & VISUALIZATION : SQL

Basic counts and distributions, new categories

```sql
# 1 - IDENTIFYING ACTIVITY SECTORS
SELECT activite, count(activite) FROM companies_info
GROUP BY activite
ORDER BY count(activite) DESC
LIMIT 10;
```

activite	count...
Programmation informatique	18
Édition de logiciels applicatifs	6
Conseil en systèmes et logiciels informatiques	5
Conseil pour les affaires et autres conseils de gestion	5
Portails Internet	5
Conseil en relations publiques et communication	3
Autres activités de soutien aux entreprises n.c.a.	2
Activités spécialisées, scientifiques et techniques diver...	2
Activités des agences de presse	2
Édition de chaînes thématiques	1

```sql
# 2 - HOW MANY EMPLOYEES DO COMPANIES HAVE
SELECT
    CASE WHEN effectif LIKE '%0 salarié%' then "0"
         WHEN effectif LIKE '%Au moins 1 salarié%'
            OR effectif LIKE '%Entre 1 et 2%'
            OR effectif LIKE '%Entre 3 et 5%' then "1 to 5"
         WHEN effectif LIKE '%Entre 6 et 9%' then "6 to 9"
         WHEN effectif LIKE '%Entre 10 et 19%' then "10 to 19"
         ELSE "20 or more"
    END AS Number_employees, count(effectif) as count
FROM companies_info
GROUP BY Number_employees
ORDER BY count(effectif) DESC;
```

Number_employees	count
0	36
1 to 5	16
10 to 19	5
20 or more	2
6 to 9	2

DATA ANALYSIS & VISUALIZATION : SQL – 2


```sql
# 3 - With these new categories, assess what type of diplomas people in direction positions have in different companies
WITH new_education AS (
SELECT ind_id,
        CASE WHEN schoolName LIKE '%Sciences Po%' OR schoolName LIKE '%IEP%'
                OR schoolName LIKE "%Institut d'Etudes Politiques%"
                then "IEP"
             WHEN schoolName LIKE '%Universi%' OR schoolName LIKE '%College%'
                then "Université"
             WHEN schoolName LIKE '%School%' OR schoolName LIKE '%ESCP%' OR schoolName LIKE '%CELSA%'
                OR schoolName LIKE '%school%' OR schoolName LIKE '%HEC%' OR schoolName LIKE '%ESSEC%'
                OR schoolName LIKE '%Management%' OR schoolName LIKE '%INSEAD%'
                then "Business school"
             WHEN schoolName LIKE '%journalism%' OR schoolName LIKE '%IFP%'
                OR schoolName LIKE '%ESJ%' OR schoolName LIKE '%CFJ%'
                then "Journalisme"
             WHEN schoolName LIKE '%Lycée%' OR schoolName LIKE '%Collège%' OR schoolName LIKE '%Prépa%'
                then "Lycée ou CPGE"
             WHEN schoolName LIKE '%EPITECH%' OR schoolName LIKE '%ENSSAT%' OR schoolName LIKE '%Télécom%'
                OR schoolName LIKE '%Polytech%' OR schoolName LIKE '%Mines%'
                then "Ecole d'ingénieur"
             ELSE "Other"
    END AS school_type
FROM people_education
)

SELECT pe.companyName, ne.school_type, count(ne.school_type)
FROM people_experience pe
LEFT JOIN new_education ne on pe.ind_id= ne.ind_id
WHERE title_direction=1
GROUP BY pe.companyName, ne.school_type
ORDER BY count(ne.school_type) desc, companyName;
```

companyName	school_type	count(ne.school_ty...
bluenove	Business school	12
cap collectif	Université	9
change.org	Université	8
change.org	Other	7
make.org	Business school	7
fluicity	Université	6
STIG	Other	6
cap collectif	Other	5
make.org	Université	5
open source politics	Other	5
abcdeep	Other	4
afup	Ecole d'ingénieur	4
civocracy	Other	4
fluicity	Business school	4
impact hub berlin	Lycée ou CPGE	4
sloop	Other	4
the one campaign	Other	4
VOXE	Other	4
VOXE	IEP	4
VOXE	Université	4
bluenove	Université	3
bluenove	Ecole d'ingénieur	3
decidim	Université	3

MODELING & PREDICTION

RECREATING THE DATASET OF PEOPLE (2 ITERATIONS)

- Concatenating different collected datasets of linkedIn profiles
- Basic cleaning : dropping duplicates, dropping columns with too many missing values (>len(dataset)/2)

NLP & RANDOM FOREST CLASSIFIER

- NLP because : text
- RFC because : classifier, tree structure, Howard Becker
- Choosing a target for prediction : industries, companies, or civic tech 1/0 ?

KEY FINDINGS

- It works on raw data - and even better with manual resampling !
- RFC quickly overfits
- Size doesn't matter (much)
- Forest depth and size do
- Gridsearch isn't always the best choice
- Ngrams offer different results
- Stop words, stemmers and TF-IDF need improvement !

	Accuracy_Score	Number_words
RFC_raw_CVec	0.571429	20683.0
RFC_raw-balanced_CVec	0.666667	20683.0
RFC_raw-balanced_CVec_BP	0.571429	20683.0
RFC_raw-balanced_CVec_BmP	0.714286	20683.0
RFC_CVec_E50_D17_nG	0.619048	90796.0
RFC_CVec_balanced_E50_D17_nG	0.619048	90796.0
RFC_CVec_E50_D17_nG_stem	0.428571	17726.0
RFC_TFIDF_E50_D17	0.428571	20683.0

MODELING & PREDICTION

MORE DETAIL ?

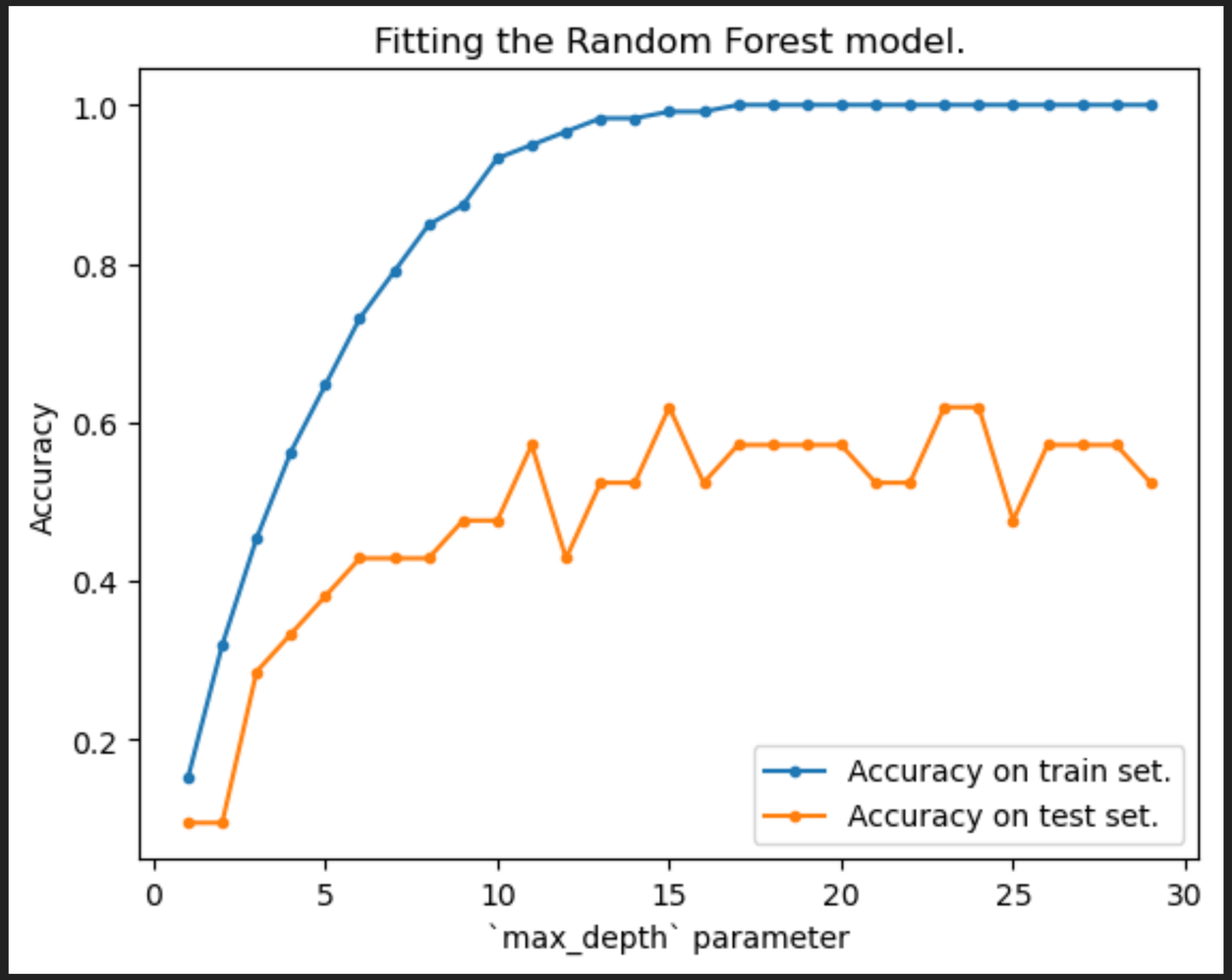
- Our first try returned an accuracy of 0.57 (for 44 categories)

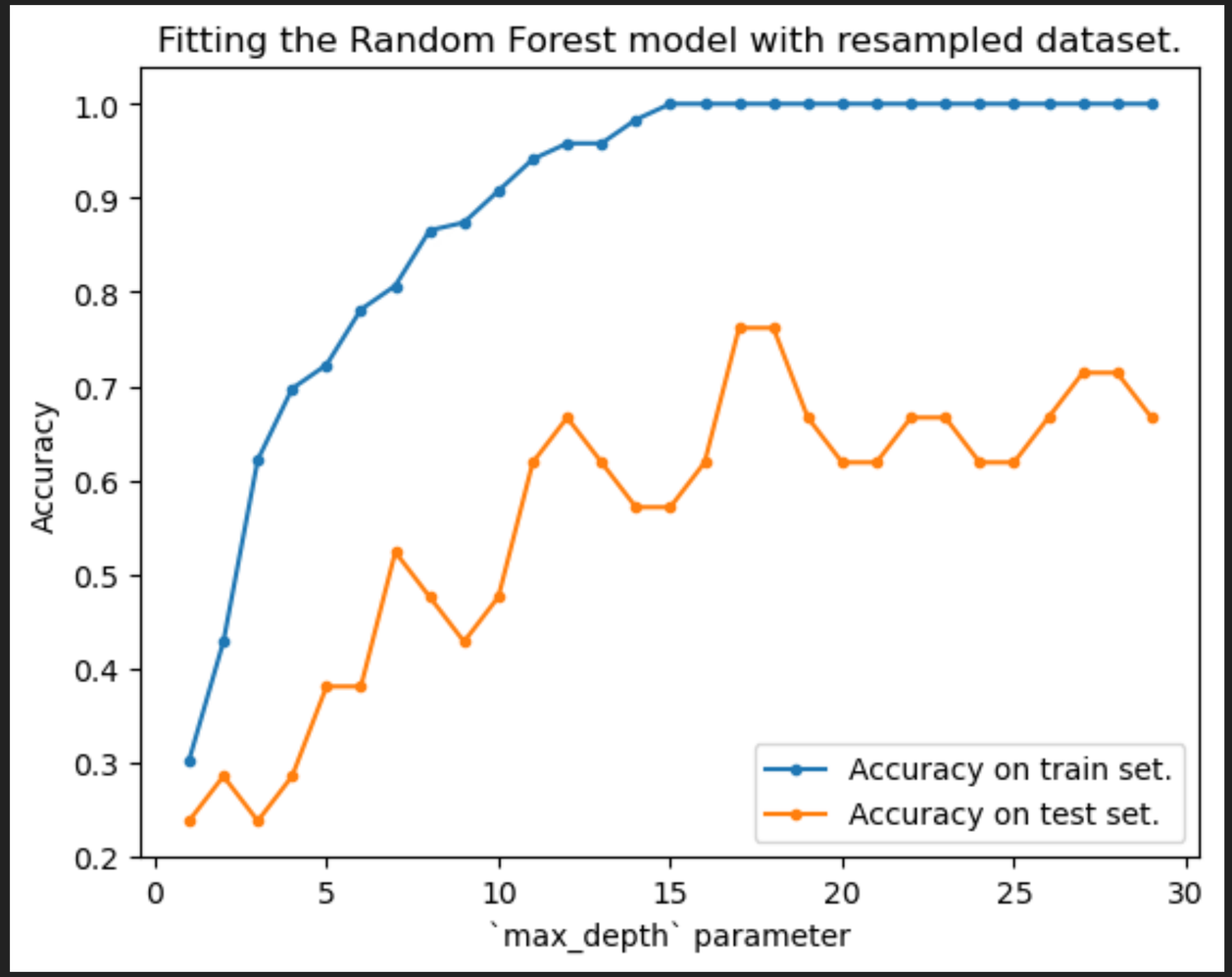
model1_confusion_matrix

Predicted industry \ Actual Industry	Civic Social Organization	Computer Software	Government Administration	Information Technology and Services	Management Consulting	Performing Arts
Civic Social Organization	**4**					
Computer Software		**2**				
Government Administration			**1**			
Information Technology and Services		**1**		**3**		
Management Consulting					**2**	
Market Research		1				
Online Media		1				
Research		1				
Architecture Planning				1		
Higher Education				1		
Information Services				1		
Investment Banking				1		
Leisure Travel Tourism						1

MODELING & PREDICTION

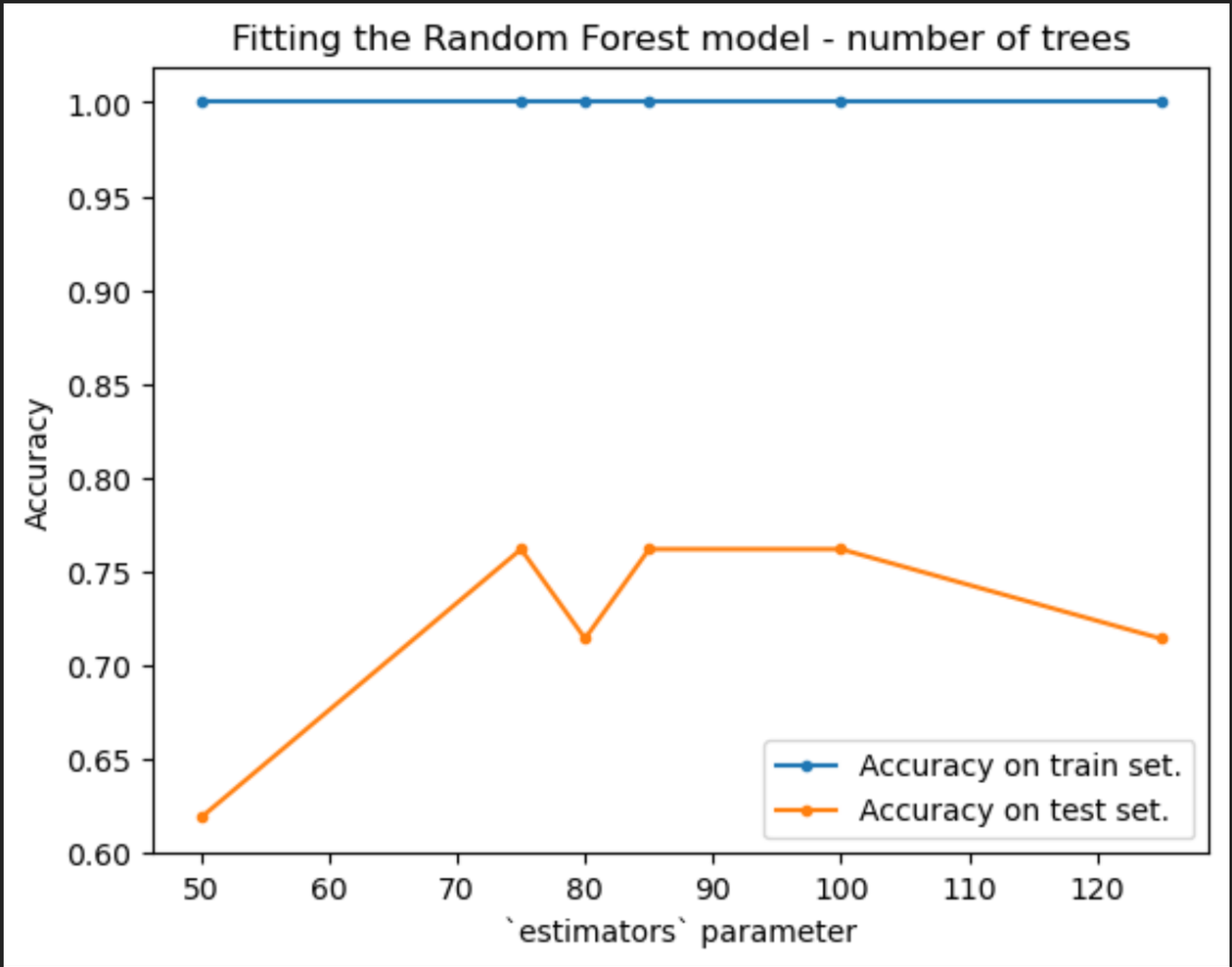
- Overfitting and manual parameter selection (max depth)





MODELING & PREDICTION

- Overfitting and parameter selection (number of trees)




```python
# If we fit it on the resampled dataset
# it gets a far better score (0.88)
# with a recommended max depth of 18 and 125 estimators.
# We can try pushing it farther
# It gets a score of 0.91 with a max_depth of 20 and 175 estimators
# although this basically means more than 1 tree per person :)

grid_search_cv = GridSearchCV(model_rf, {
                              'n_estimators':[110, 125, 150, 175],
                              'max_depth': [17, 18, 19, 20, 21]
                              },
                              cv=3, scoring='accuracy')
grid_search_cv.fit(X_train_rs, y_train_resampled)
print(grid_search_cv.best_score_)
print(grid_search_cv.best_params_)
```

✓ 1m 1.9s

```
0.9159621936476535
{'max_depth': 20, 'n_estimators': 175}
```

WILL AI REPLACE SOCIAL SCIENTISTS ?

- Probably not 😅
- When adding complexity (languages, projects, publications (i.e. columns with missing data), the model has trouble following - we get to .22 with a lot of tweaking and a lot of trees (400+)...
- Still better than 1/47

```
1  model_rf = RandomForestClassifier(random_state=42, class_weight='balanced_subsample')
2  model_rf.fit(X_train, Y_train)
```

✓ 0.1s

```
RandomForestClassifier
RandomForestClassifier(class_weight='balanced_subsample', random_state=42)
```

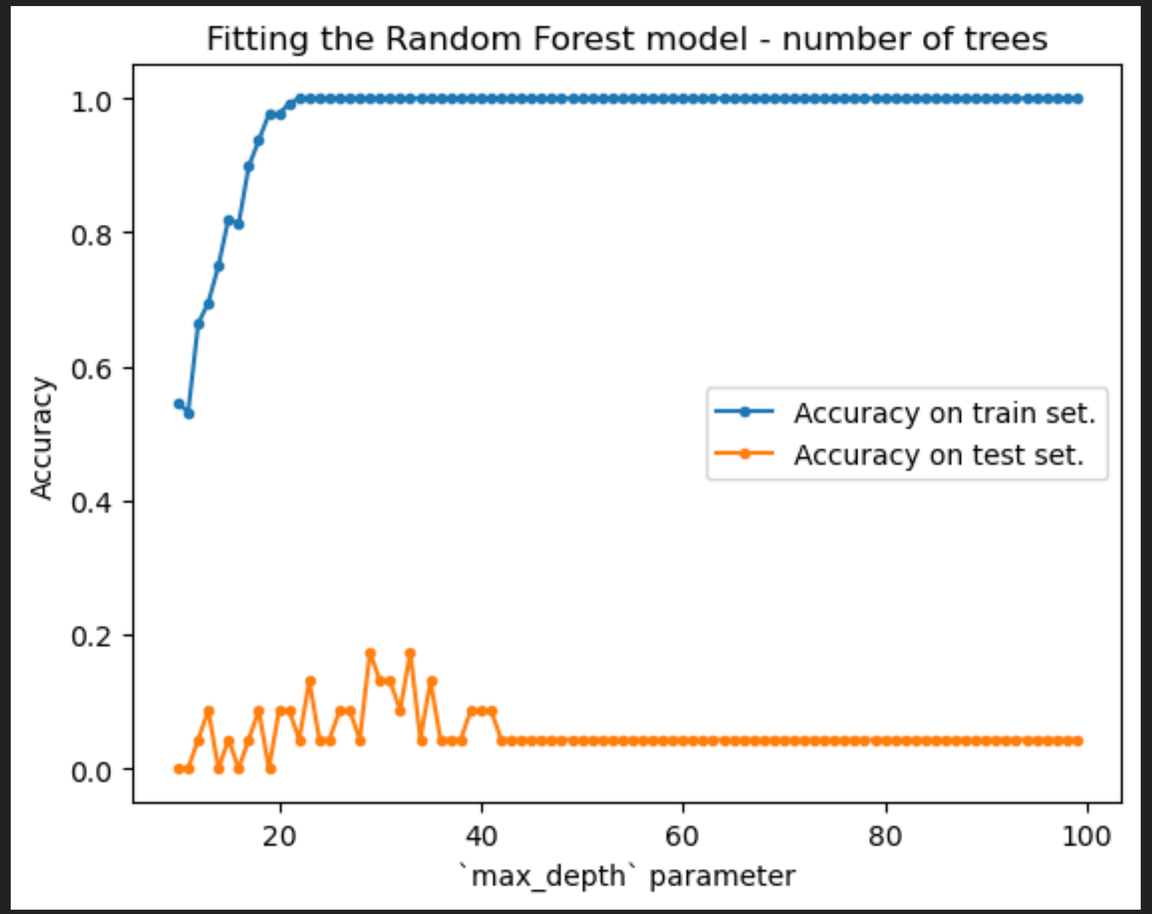
```
1  y_pred = model_rf.predict(X_test)
```

✓ 0.0s

```
1  accuracy_score(Y_test, y_pred)
2  # IS NOT GREAT :)
```

✓ 0.0s

```
0.17391304347826086
```



FUTURE IMPROVEMENTS FOR PREDICTION

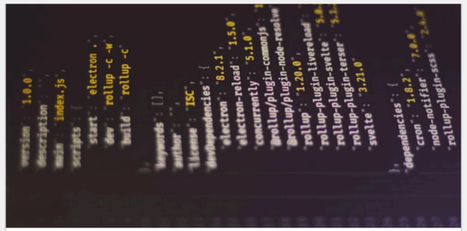
- A 'next company' recommender based on nearest neighbors
- Using a PCA to make clusters of people
  or MCA, actually, for correlated features
- Use PCA to make clusters of companies
- Change the y to
  - civic tech yes/no
  - or to the clusters

NEXT STEPS

Trello Workspaces Recent Starred Templates Create Search TF

Civic_tech_whos_who Workspace visible Board Power-Ups Automation Filter TF Share

Data collection



Online data on individuals in the civic tech sector in France

5/5

Online data on organizations in the civic tech sector in France

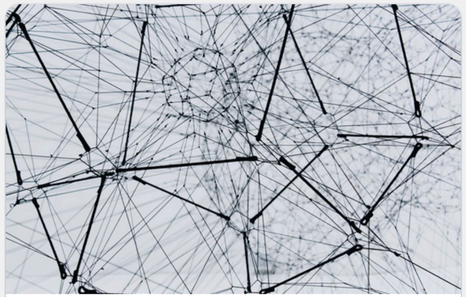
3/3

Add a card

Data analysis

Database construction & data cleaning

15/15



Data analysis : Python EDA, SQL

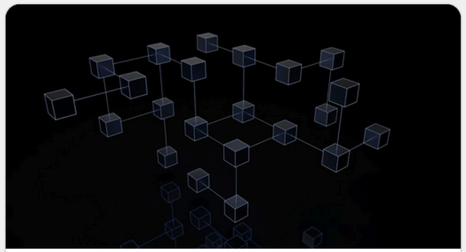
9/9

Add a card

Visualization & modeling

Data visualisation

2/2



Data analysis and modeling

6/6

Add a card

Presentation

Slides and github repo

1 2/2

Report writing

Add a card

Bonuses

Beautify code and repo

Improve LinkendIn scraper to get previous experience

R Selenium code for company data scraping

Additional companies in database

Fix/ get around LinkedIn restriction

Categorical encoding for education information

Statistical description of individuals and companies (// ecosystems)

Add a card

- Thank you !

- Notebooks & readme available here : https://github.com/tatdef/IH_final_project

- Comments welcome !