UNIVERSITÉ RENNES 2

Rapport de projet 2020

Création d'un site web composé de cartes interactives à partir d'une base de donnée

---

Groupe : Clément CAILLARD, Mélanie GUILLOUET, Pauline HAMON-GIRAUD et Rémi LEDUC

Encadrant : Romain TAVENARD

SOMMAIRE

1. Introduction

Le choix du ter s'est fait en plusieurs étapes. Pour commencer notre groupe de TER s'est composé de personnes ayant déjà travaillé ensemble dans le passé ce qui simplifie la communication.

Pour le choix du sujet, nous voulions un sujet comprenant de l'informatique car c'est une matière à laquelle les membres du groupe portent de l'intérêt. Au vu de la vitesse d'appropriation des sujets informatiques proposés, nous avons finalement choisi de proposer notre sujet. Ce choix nous permet d'être un peu plus libre sur la problématique, ce qui nous motive davantage. Nous voulions faire un choix de sujet qui nous permettrait de découvrir de nouveaux aspects des langages de programmation que nous connaissons actuellement.

Afin de définir plus précisément les objectifs et la nature du projet, nous avons commencé par procéder à la recherche de bases de données, notamment sur le site data.gouv. Plusieurs sujets nous ont paru intéressants au premier abord, notamment une base de données fonctionnant par une API, sur les événements (festivals, concerts, conférences …) en France. L'idée nous est venue de créer un carte interactive accessible par navigateur à l'aide de cette base de données. Cela qui nous paraissait intéressant par la multitude de langages de programmation et de compétences nécessaires pour la réalisation d'un tel projet.

Nous avons également considéré la réalisation d'une carte interactive des voyages ferroviaires en France, grâce à une base de données de la SNCF libre d'accès mais elle présentait plusieurs contraintes : l'utilisation d'une librairie spécifique et une faible quantité d'informations sur les voyages. Nous sommes donc revenus sur notre choix de départ des événements en France.

Suite à une discussion avec notre encadrant, Romain Tavenard, nous avons décidé également d'ajouter à notre site web une page contenant des graphiques et différentes statistiques à propos des évènements de la base de données en utilisant la librairie Bokeh. Cela en plus de la carte qui reste l'élément principal du projet.

2. Partie organisationnelle

2.1. Organisation et planification

Notre organisation de travail s'est passée de deux manières différentes, la première partie s'est passée de janvier à mi mars puis de mi mars à mi avril dû à la fermeture de l'université liée à la crise sanitaire.

Durant les deux premières semaines nous nous sommes regroupés chaque jeudi après midi pour définir notre projet, ce que nous avions envie de réaliser et le rendu final que nous souhaitions obtenir. A cet instant, nous avions pour but de faire un site de cartographie dynamique. Nous avons eu ensuite une réunion avec Romain Tavenard pour lui présenter nos objectifs. Il nous a alors conseillé de découvrir certaines librairies et proposé d'ajouter des graphiques à notre site. Après avoir fait mûrir nos idées et défini ensemble nos objectifs, nous avons choisi notre base de données. Ensuite nous nous sommes penchés ensemble, et de façon individuelle, sur la découverte des librairies python et des différentes possibilités pour parvenir à nos objectifs finaux. Il nous paraissait essentiel que chacun puisse comprendre le fonctionnement des différents modules pour la suite du projet. Pour partager nos idées, nous avons créé un document Google drive où nous avons partagé nos connaissances, que ce soient des idées, des objectifs ou propriétés sur les librairies. Mi février nous avons eu une deuxième réunion avec notre professeur référent pour lui présenter nos avancés et comment nous souhaitions procéder pour la suite du projet. Nous nous sommes alors rendu compte que nous avions pris une semaine de retard par rapport à notre planning prévisionnel.

Une fois la partie découverte et d'apprentissage des modules terminée nous avons découpé le travail en deux parties, deux personnes ont commencé à développer le code concernant la cartographie et deux autres se sont intéressés au code sur la partie graphique. Pour partager nos codes nous avons utilisé la plateforme github où grâce à un espace de travail commun tous les membres du groupe avaient accès aux programmes des autres. Notre troisième réunion avec Romain Tavenard n'ayant pas pu avoir lieu à cause de la fermeture de la fac nous lui avons partagé notre avancé par mail. Grâce à notre choix d'avoir divisé le travail nous avons réussi à rattraper notre retard sur la programmation de notre projet.

La crise sanitaire empêchant de travailler à l'université nous avons procédé par une autre forme d'organisation pour la suite du TER. Les premières semaines il

a fallu se mettre d'accord sur une nouvelles organisation prenant en compte les moyens à dispositions et les contraintes de chacun. Nous avons donc décidé de se retrouver par le biais de réunions en visioconférence. Ces réunion nous permettaient d'expliquer nos avancés, nos problèmes et de se fixer de nouveaux objectifs pour la semaine suivante. Nous avons continué pendant cette période d'utiliser github et google drive pour partager nos avancés. L'ensemble du groupe n'ayant pas accès à tous les moyens techniques nécessaires, il était plus difficile de se diviser le travail et d'atteindre les objectifs en temps voulu. Nous avons donc accumulé environs deux semaines de retard. Cependant certains ont pu développer plus rapidement leur partie et ont ainsi permis au projet de rattraper son retard. Ainsi malgré le retard accumulé pendant la période du confinement, nous avons réussi à atteindre nos objectifs pour la date de rendu.

2.2. Analyse critique

Durant ce projet nous avons rencontré plusieurs décalages, en effet notre première difficultée a été de définir nos objectifs. Nous pensions que de travailler sur notre propre sujet de TER allait être un avantage dans l'organisation de notre travail, cependant cela a été notre principale difficultée. En effet nous n'avions aucun objectifs imposés et aucune direction pour nous aider à définir des tâches et ainsi répartir notre travail.

De plus nous nous sommes aperçu que nous avons peu fait appel à notre professeur référent, ce qui nous a sûrement desservie dans notre organisation puisque c'est une aide considérable dans la construction d'un projet. Nous avons donc travaillé essentiellement en autonomie et surement passé plus de temps à résoudre des problèmes techniques.

Ensuite nous avons dû faire face aux difficultés durant le confinement, en effet il était plus compliqué de travailler en groupe sans avoir la possibilité de se réunir. Chacun ayant des contraintes différentes nous avons dû nous accorder. Fixer une organisation stricte avec des objectifs à atteindre chaque jeudi n'était pas tenable, certains ayant des difficultés techniques. De plus il nous était plus difficile de s'entraider à distance. Grâce aux facilités en programmation de certains nous avons pu continuer notre progression, cependant cela a engendré une quantité de travail inégale entre les étudiants.

3. Partie technique

3.1. Sujet et appropriation

Comme expliqué précédemment cette partie du projet nous a pris plus de temps que nous le pensions. En effet nous avons formé notre groupe dans le but de créer un site de cartographie dynamique, cependant définir notre sujet de façon détaillé nous à été plus difficile, cela demande du temps et beaucoup de communication. Après plusieurs jours de réflexion nous y sommes parvenu et nous avons commencé à réfléchir sur les étapes de construction de notre projet. Cependant il nous a été compliqué de définir ces étapes. Il a tout d'abord fallu se mettre d'accord sur la forme précise du projet que nous souhaitions obtenir, tout en pensant au aspect technique que cela engendrait. Mais il est compliqué de définir des objectifs sans se rendre compte du temps et de la difficultée que cela inclus. De plus nous n'arrivions pas à nous rendre compte du niveau de difficulté attendu dans un projet de TER. Ensuite nous avons dû choisir la base de donnée avec laquelle nous voulions travailler. Nous avons fait des réunions et hésité entre plusieurs bases de données. Pour le choix de la base de données il fallait une api qui ne demandait pas de clé d'authentification et qui fournisse des données géographiques. Nous avons donc choisi une base de données avec un thème que tout le monde apprécient: les événements en france et qui correspond à nos besoins techniques. Ainsi le temps de recherche, de réflexion et de discussion sur nos objectifs nous a permis de nous approprier notre sujet.

3.2. Travail effectué

Site web :

Nous avons commencé par s'inspirer sur d'autres sites web pour voir les possibilités.

Il a fallu créer la page web en incrustant une carte et des graphiques. Pour cela nous avons généré une carte (avec folium) vierge et des graphes (avec bokeh). Il a donc fallu chercher comment pouvoir exporter la carte et les graphiques dans une chaîne de caractère intégrable au code HTML du site. Après des recherches dans les docs des librairies respectives, nous avons trouvé comment obtenir le code html sous forme de chaîne de caractère.

Pour folium le code consiste en 2 et un <iframe> . La carte prend 100 % du bloc ou elle est mise. La carte s'adapte selon la taille de la fenêtre tout en gardant son ratio d'origine.

Pour bokeh le code consiste en une balise <script> qui contient le graphe et une balise qui permet de placer le graphe dans le code de la page. Les tailles du graphe ne sont, de base, pas adaptables et on ne peut pas changer avec le script CSS. Il est toutefois possible, lors de la création du graphe dans python, de définir la taille de celui-ci et de pouvoir lui demander de s'adapter selon la taille de la fenêtre active, en gardant ou non son ratio d'origine.

Après avoir réussi à obtenir notre carte et nos graphes, il a fallu créer la page web. Même avec des bonnes bases grâce au cours d'HTML et de CSS, il est tout de même plutôt compliqué de créer un site web depuis une page vierge. Nous avons défini une structure au sites que vous pouvez retrouver dans le rapport. Finalement le code du site web nous aura pris plus de temps que prévu dans le planning car il est assez complexe à distance de coder à plusieurs un site mais le résultat est plutôt convaincant à notre goût.

Le dataframe:

Pour la création de nos représentations graphiques, il nous a fallu importer des données. Nous avons, comme décidé précédemment, utilisé l'api "Événements Publics - OpenAgenda" de data.gouv.fr. Pour importer ces données, nous avons utilisé la librairie urllib.request. Nous voulions au départ voir les événements de l'année en cours mais la limite de l'api étant de 10 000 événements, nous avons donc choisi d'importer les événements du mois en cours.

Les données récupérées par l'api sont dans un format json, après avoir compris comment était composé le json grâce à plusieurs test, nous avons stocké les données dans un dataframe. Pour ce faire nous avons utilisé la librairies Pandas qui était claire et assez simple d'utilisation. Elle comprenait également une fonction permettant de faire un dataframe à partir d'un json ce qui nous a simplifié la tâche.

Nous sommes ensuite passés au nettoyage des données pour avoir le dataframe le plus propres possible. Nous avons commencé par supprimer les colonnes inutiles pour optimiser le dataframe. Ensuite nous avons décidé de garder uniquement les événements situés en France. Puis il a fallu supprimer les événements ne contenant pas de titre ou de coordonnées gps (latitude, longitude) car ils n'étaient pas utilisables pour la carte que nous réaliserons plus tard. Vient ensuite la suppression

des données en double, pour être sûr qu'une information soit en double, nous avons supprimé les événements qui avaient le même titre et la même date de début.

Carte:

Pour la carte nous nous étions beaucoup exercés avant de nous lancer dans la réalisation du TER ce qui nous a beaucoup aidé à être efficace car nous savions où nous allions. Nous avons tout d'abord créé une carte vierge centrée et zoomée sur la France sans barre d'échelle. Puis nous avons créé un groupe de points réunissant 5 sous groupes de points qui sont triés en fonction de la date de début de l'événement.

Pour ajouter les points sur la carte il a fallu boucler sur les lignes du dataframe, nous utiliserons cette boucle pour les graphes également, pour éviter de faire trop de boucles qui ralentiraient le programme.

Chaque ligne du data frame correspond à un événement. Dans la boucle pour chaque événement nous créons un point bleu avec un logo "i" de info à l'intérieur. Nous avons réglé le popup (page html qui s'ouvre sur la carte lorsque l'on clique sur le point) pour qu'il affiche tout en haut côtes à côtes : l'image de l'événement et la date de celui ci si l'image existe, sinon juste la date. La date étant composée soit de la date début et de fin si elle sont différentes ou seulement la date de début si les 2 dates sont identiques. Puis en dessous, le titre de l'événement auquel on a intégré le lien du site de l'événement s'ouvrant dans un onglet à part. Puis dans l'ordre, s'ils existent, la description de l'événement, les infos sur le prix et des infos sur le lieux et le moment.

Nous avons ensuite intégré l'événement dans le sous groupe de point qui lui correspond en fonction de sa date de début. Après avoir fais de même pour tous les événements, nous rajoutons les groupes de points à la carte et nous mettons le code html de celle-ci dans une variable que nous utiliserons plus tard.

Les graphes :

Pour le graphes qui compte le nombre d'événements par région, nous avons créé un dictionnaire avec les 13 régions pour clé et des 0 en valeur. Dans la boucle précédemment créée, on compare la région de l'événement à celles du dictionnaire et on ajoute 1 à la valeur de la région qui correspond.

En sorti de boucle on ajoute les données dans une base spécifique à bokeh (ColumnDatasource) pour simplifier la création du graphe. Puis nous créons la figure

(boîte) qui accueillera le graphe. On met la hauteur définie pour le site ainsi qu'une largeur adaptable, on enlève la boîte à outil qui serait inutile dans le cas présent, une légère marge définie pour le site et un popup qui affiche le nombres d'événements lorsqu'on survole les données à la souris.

Vient ensuite l'ajout des données dans la figure qui se fait assez logiquement sous la forme de barres. En abscisse il y a les régions et en ordonnée le nombre d'événements. Chaque région a sa propre couleur définie aléatoirement grâce à une option de Bokeh. Nous avons changé l'orientation du nom des régions pour plus de lisibilité.

Pour le graphe qui compte le nombre d'événements selon le jour du mois le principe est relativement le même sauf que, dans le dictionnaire de départ, on met en clé les jours du mois selon le nombre de jours dans le mois actuel. On a décidé de tracer une ligne plutôt que des barres car représenter 30 barres est peut lisible. En abscisse du graphe on a les jours du mois et en ordonné le nombre d'événement.

Nous avons ensuite stocké le code html des graphes dans des variables qu'on utilisera ensuite.

La création de la page web à partir de python :

Pour la création de la page web nous avons d'abord créé un chaine de caractère contenant tout le code HTML en y intégrant les variables créée précédemment aux endroits nécessaires.

Nous avons ensuite utilisé la fonction os.mkr qui crée un dossier dans le répertoire, nous avons mis une exception si le dossier existe déjà. Puis grâce à la fonction open directement intégrée dans python nous ouvrons le fichier "index.html" en mode "w" qui le crée si il n'existe pas. Nous y intégrons notre code html et nous fermons le fichier.

Nous faisons de même avec le css, ce qui permet d'avoir le site web entièrement crée grâce au programme python.

3.3. Difficultés et solutions

Pour le site web :

- La plus grande difficulté fut de pouvoir intégrer la carte et les graphes de la manière souhaitée. Pour régler ce problème il a été nécessaire de faire beaucoup de recherches sur le web et de tester sur machine pour comprendre les librairies folium et bokeh et comment les intégrer à du code HTML.

- Le fait de ne pas savoir exactement le rendu final à obtenir et de partir de zéro est déroutant et change aussi beaucoup des cours de HTML/CSS qu'on peut avoir, mais se fixer des objectifs et une structure pour le site permet déjà de mieux s'y retrouver.

- La répartition des tâches pour l'élaboration d'un site web en groupe et à distance est plutôt quelque chose de complexe, cela nécessite beaucoup de communication au sein du groupe et que tout le monde puisse réaliser ses tâches à temps.

Pour l'importation et le nettoyage des données:

- Il a fallu importer uniquement les données qu'on souhaite en fonction du mois et de l'année où l'on se situe. Pour cela il a fallu comprendre comment fonctionne l'url de l'api et d'y ajouter au bon endroit le mois et l'année. Nous avons donc dû décomposer l'url et faire quelques tests pour comprendre comment celui ci fonctionne et nous avons ajouté le mois et l'année sous forme de str récupérés grâce à la librairie datetime vue en cours.

- Le nombre de données récupérées étant important (environ 6 000), le programme prend du temps à se réaliser, ce qui nous fait perdre beaucoup de temps. Pour palier à ce problèmes nous avons limité le nombre de données récupérées grâce à une information dans l'url. Il ne nous resterait plus qu'à tester les données complètes à la fin.

- Ayant décidé de travailler uniquement sur les événements ayant lieu en France, il a fallu supprimer tous les autres de notre dataframe. N'ayant pas de colonne "pays", nous avons donc fait un dataframe qui garde l'événement si la région de celui-ci est dans une liste des 13 régions de France. Cela a pour effet de supprimer les autres événements mais aussi ceux dont la région est mal orthographiée (Région loire au lieu de Centre-Val de Loire par exemple) mais ceux-ci étant en nombres très réduit nous avons décidé de les ignorer.

Pour la carte:

- Le nombre d'événements en France étant conséquent, la carte devient vite illisible si on ne zoome pas dessus. Pour palier à cette difficulté nous avons fait des recherches et nous avons finalement trouvé le système de cluster qui réunit les points en un seul avec le nombre de point que celui-ci contient lorsque il y a plusieurs points au même endroit.

- Le fait de vouloir trier les points selon leur date de début en 5 groupes pose le problème de connaitre le nombre de jours du mois en cours. Pour cela nous avons fait une courte fonction qui permet de calculer le nombre de jours du mois.

Pour les graphes:

- La plus grosse difficulté lors de la création des graphes se situe dans l'aspect visuel de celui ci et donc dans la création de la boite qui va le contenir. Il a été nécessaire de bien étudier la doc de Bokeh pour pouvoir mettre des options optimales pour nos graphes.

- La seconde difficulté, c'est que lorsqu'on souhaite aller un peu plus loin dans l'interactivité des graphes, il est nécessaire de comprendre le javascript ce qui n'est pas notre cas malheureusement (un point d'amélioration).

Pour la création de la page web à partir de python:

- On peut vite se perdre dans le code car il n'est pas mis en évidence comme sur notepad++ il est donc nécessaire d'être rigoureux.

- Le fait de tout créer à partir de python les images qui sont normalement déjà téléchargées ne peuvent pas être intégrées au nouveau. On a donc mis un chemin url vers la page de téléchargement de ces images.

3.4. Analyse critique des résultats

Nous sommes satisfaits du résultat obtenu, il y a quelques défauts évidents que nous n'avons pas pu résoudre, comme par exemple, la forme du site web sur mobile (nous avons pas pu tester) qui doit être illisible du fait des graphes qui sont en pixel et du header qui ne s'adapte pas. Nous avons malgré tout réalisé nos objectifs et découvert d'autres qui pourraient aider à l'amélioration du projet dans le futur. Nous pouvons citer comme objectif découvert : l'utilisation de serveur web grâce à des librairies python, ce qui permettrait d'actualiser le site sans devoir relancer le programme manuellement et rendre le site dynamique. On peut aussi citer le fait de mieux comprendre le javascript pour faire des graphiques plus interactifs.

4. Conclusion

Pour conclure, d'un point de vue technique, ce projet nous a permis d'enrichir nos connaissances en python en découvrant de nouvelles librairies et donc de nouvelles fonctionnalités. Cela permet d'élargir les possibilités d'utilisation du langage et nous permettra dans le futur, de pouvoir, si il est nécessaire, utiliser des data frames, créer des cartes, des graphiques et générer une page Web.

D'un point de vue organisationnel, ce travail nous a aussi permis d'apprendre à s'organiser en groupe pour un projet sur une longue durée. Il a donc fallu planifier au mieux la gestion du temps et des différentes phases. Mais il a fallu aussi diviser les tâches au sein des membres du groupe tout en permettant à chacun de pouvoir suivre et comprendre l'avancée générale du projet.
Nous en avons aussi retiré le fait que la gestion du temps devait prendre en compte différents scénarios possibles. A ce sujet, nous n'avions pas assez envisagé de temps pour la gestion des imprévus. Mais nous avons pu nous adapter et nous réorganiser.

Sur les deux points de vue précédents, nous avons pu percevoir des pistes d'amélioration pour des projets futurs du même type. Comme par exemple créer un cahier des charges et un planning mieux défini dès le départ. Mais aussi, pousser plus loin l'apprentissage des langages de programmation les plus usuels pour pouvoir élargir davantage les possibilités dans cette discipline.

5. Annexes

5.1. Exploration des librairies

Folium (création de carte)

Sources :

Documentation de folium :
https://python-visualization.github.io/folium/modules.html

Tutoriel sur les Markercluster avec folium publié par Bob Haffner :
https://medium.com/@bobhaffner/folium-markerclusters-and-fastmarkerclusters-1e03b01cb7b1

Module folium : `folium.folium`

Création d'une carte "m"			
objet folium "Map" `ex : m = folium.Map()`			
Paramètre	**Type**	**Défaut**	**Détail**
`location`	Liste ou tuple(de taille 2)	None	[latitude, longitude]
`tiles`	Str	OpenStreetMap	Type de carte utilisée
`zoom_start`	Int	10	Zoom de départ
`min_zoom`	Int	0	Zoom minimal autorisé
`max_zoom`	Int	18	Zoom maximal autorisé

Module map : `folium.map`

Ajout d'un marqueur à la carte "m"
objet map "Marker"

`ex : folium.Marker().add_to(m)(methode add_to())`			
Paramètre	**Type**	**Défaut**	**Détail**
`location`	Liste ou tuple (de taille 2)	None	[latitude, longitude]
`popup`	Str ou folium.Popup	None	Légende du marqueur
`icon`	folium.Icon		Type d'icône du marqueur

Paramètre de l'icone du Marker			
objet map "Icon" `ex : folium.Marker(icon=folium.Icon())`			
Paramètre	**Type**	**Défaut**	**Détail**
`color`	str	"blue"	Couleur du marker
`icon_color`	str	"white"	Couleur de l'icône sur le marker
`icon`	str	"info-sign"	Type d'icône du marqueur https://fontawesome.com/icons
`prefix`	str	"glyphicon"	Utiliser "fa" pour utiliser les icones de fontawesome "glyphicon" pour bootstrap 3
`angle`	int	0	Degré de rotation de l'icône

Paramètre du popup du Marker			
objet map "Popup" `ex : folium.Marker(popup=folium.Popup())`			
Paramètre	**Type**	**Défaut**	**Détail**
`html`	str		Contenant du popup
`parse_html`	bool	False	
`max_width`	int (pourcentage de pixels)	100(%)	Largeur maximale du popup
`show`	bool	False	True : Popup affiché au chargement de la carte

Module Vector Layers : `folium.vector_layers`

Ajout d'un cercle autour d'un point à la carte "m"			
objet vector_layers "Circle" `folium.Circle().add_to(m)`			
Paramètre	**Type**	**Défaut**	**Détail**
`location`	Liste (de taille 2)		[latitude, longitude]
`radius`	Float		Rayon du cercle
`popup`	Str ou folium.Popup		Légende du marqueur
`fill`	Bool	False	Remplissage du cercle
`fill_color`	Str	color	Couleur du remplissage

Ajout d'une couche de contrôle			
`folium.LayerControl()`			
Paramètre	**Type**	**Défaut**	**Détail**
position	string	'topright'	position de la couche : 'topleft', 'topright', 'bottomleft' ou 'bottomright
collapsed	boolean	true	

Création de groupe de points affichable			
`folium.FeatureGroup()`			
Paramètre	**Type**	**Défaut**	**Détail**
name	string		Nom du groupe

Création de sous groupe de points affichable			
`folium.plugins.FeatureGroupSubGroup()`			
Paramètre	**Type**	**Défaut**	**Détail**

group	Layer		Nom du groupe associé
name	string		Nom du sous groupe créé

Il existe aussi d'autres classes permettant de tracer des polygones, rectangles, etc... Mais qui ne nous semblent pas utiles pour notre projet.

Ajout d'un cluster (groupe de points sous forme d'un seul)			
class : folium.plugins `MarkerCluster()`			
Paramètre	**Type**	**Défaut**	**Détail**
`locations`	list	None	Liste de liste de taille 2 [lat, lon]
`popups`	list	None	Liste de popup
`icons`	list	None	Liste d'icônes
`name`	str	None	Nom du Layer

Pandas (utilisation d'une BDD)

Sources :

Documentation de Pandas :

https://pandas.pydata.org/docs/

Site python-simple.com, partie sur les data frames :

http://python-simple.com/python-pandas/dataframes-indexation.php

Création d'un DataFrame"df"		
`df = pandas.DataFrame()`		
Paramètre	**Type**	**Détail**
`index`	list	liste pour les lignes
`columns`	list	liste pour les colonnes

méthodes	**Détail**
`describe()`	bref résumé statistiques
`iloc[]`	sélectionner une partie du df
`head()`	affiche n premières lignes
`tail()`	affiche n dernières lignes
`rename()`	Renommer les colonnes du df

Avantages:

gérer les données de manière optimale.
Pas de soucis de conversion d'un fichier csv au contenu mixte

Bokeh

Sources :

Documentation de Bokeh :

https://docs.bokeh.org/en/latest/docs/user_guide/quickstart.html

Création d'un graphe "p"		
`p = figure()`		
Paramètre	**Type**	**Détail**
`plot_width`	numeric	largeur du graphe
`plot_height`	numeric	hauteur du graphe
`tools`	list	outils disponible pour l'utilisateur du graphe
`legend_label`	str	légende du graphe
`title`	str	titre du graphe
`sizing_mode`	str	taille adaptable ou non

méthodes	**Détail**
`line()`	graphe lignes (entre points)
`vbar()`	histogramme vertical

`hbar()`	histogramme horizontal
`wedge()`	diagramme circulaire

Intégration d'un plot Bokeh dans une page HTML :

Méthodes	**Détail**
`bokeh.embed.components()`	renvoie un tuple (script,div) str div et script à intégrer dans le code html
Attention : il faut rajouter cette ligne dans le header de la page html pour que ça fonctionne: "<script type="text/javascript" src="https://cdn.bokeh.org/bokeh/release/bokeh-2.0.2.min.js" integrity="sha384-ufR9RFnRs6lniiaFvtJziE0YeidtAgBRH6ux2oUItHw5WTvE1zuk9uzhUU/FJXDp" crossorigin="anonymous"></script>"	

Avantages:

bonne intégration avec pandas
Intéractif

Geocoding

Utilisation de geocoder pour obtenir des coordonnées gps à partir d'une chaine de caractère

Méthodes	**Détail**
`g= geocoder.osm("adress")`	créer un objet geocode ici g
`g.latlong`	renvoi la latitude et la longitude de g sous la forme [lat, long]

5.2. Base de données

La base de données utilisée est extraite de : opendatasoft.com
Il s'agit de la liste des événements publics publiés dans les agendas OpenAgenda.
Identifiant du jeu de données : `evenements-publics-cibul`

Disponible avec le lien suivant :
https://public.opendatasoft.com/explore/dataset/evenements-publics-cibul/information/?disjunctive.tags&disjunctive.placename&disjunctive.city&sort=date_start&dataChart=eyJxdWVyaWVzIjpbeyJjb25maWciOnsiZGF0YXNldCI6ImV2ZW5lbWVudHMtcHVibGljcy1jaWJ1bCIsIm9wdGlvbnMiOnsiZGlzanVuY3RpdmUudGFncyI6dHJ1ZSwiZGlzanVuY3RpdmUucGxhY2VuYW1lIjp0cnVlLCJkaXNqdW5jdGl2ZS5jaXR5Ijp0cnVlLCJzb3J0IjoiZGF0ZV9zdGFydCIsImxvY2F0aW9uIjoiNiw0NS43NzUxOSwzLjE4NjA0IiwiYmFzZW1hcCI6Im1hcHF1ZXN0In19LCJjaGFydHMiOlt7ImFsaWduTW9udGgiOnRydWUsInR5cGUiOiJsaW5lIiwiZnVuYyI6IkNPVU5UIiwic2NpZW50aWZpY0Rpc3BsYXkiOnRydWUsImNvbG9yIjoiI0ZGNTE1QSJ9XSwieEF4aXMiOiJ1cGRhdGVkX2F0IiwibWF4cG9pbnRzIjoiIiwidGltZXNjYWxlIjoieWVhciIsInNvcnQiOiIifV0sImRpc3BsYXlMZWdlbmQiOnRydWUsImFsaWduTW9udGgiOnRydWV9&calendarview=month&location=6,45.77519,3.18604&basemap=mapquest

Nous l'avons utilisée via l'API proposée.

5.3. Code

Le code pythons se constitue d'un seul fichier .py qui sera joint au rapport à part.
L'exécuter créer un dossier contenant la page d'accueil en .html et sa feuille de style .css associée.

Description du rendu final de la page Web

1 - Page d'accueil

<table>
<tr><td colspan="3">Logo Titre (Header fixe) Menu</td></tr>
<tr><td colspan="3">La base de données en quelques chiffres</td></tr>
<tr><td>Texte</td><td colspan="2">La carte</td></tr>
<tr><td colspan="2">Graphe 1</td><td>Texte</td></tr>
<tr><td>Texte</td><td colspan="2">Graphe 2</td></tr>
<tr><td colspan="3">Footer</td></tr>
</table>