
Smart data : analyses, valorisations et visualisations de données stratégiques multi sources



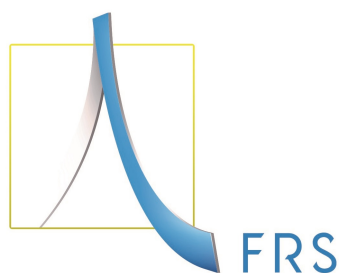
RAPPORT DE STAGE
MASTER MIAGE I - PROMOTION 2018-2019

9 Avril 2017 — 31 Juillet 2017

Auteur : Alexandre PETIT-PAS

Tuteur : Gaétan LE CHAT

Référent : Pascal POIZAT



Remerciements

Je tiens à remercier Gaétan Le Chat, mon tuteur de stage, qui m'a aidé et accompagné pendant toute la durée du stage.

Je tiens à saluer Cheikhouna Diouck, stagiaire en école d'ingénieur et collègue stagiaire du pôle R&D avec qui j'ai pu partager de bons moments. On doit s'améliorer sur les pronostics des matchs de football par contre !

J'aimerais aussi saluer Kymble Christophe qui m'a apporté un point de vue nouveau et un avis toujours franc sur le travail que je faisais. Cela m'a permis d'avoir constamment un retour critique sur mon travail.

Un grand remerciement à Vincent Grollemund qui m'a beaucoup aidé à mes débuts de D3 et que j'ai pu aider pour des problèmes de javascript. C'est un langage particulier mais on peut faire de belles choses avec !

Un grand merci à toutes les personnes de FRS qui m'ont chaleureusement accueilli et avec qui j'ai passé de bons moments durant le stage.

J'aimerais aussi mentionner "les bébés minions", des étudiants de L3 en détresse qui ont repris l'outil fait dans mon précédent stage et que j'ai pu aider. N'oubliez pas, une nouvelle fonctionnalité est à faire pour hier.

Une pensée à mes camarades de promotion avec qui nous avons partagé nos différentes expériences.

Enfin je tiens à remercier François Delbot, sans qui je n'aurais sûrement pu réaliser ce stage.

1	Introduction	6
2	FRS Consulting	8
3	Recommandation de dispositifs d'aide à l'innovation	10
3.1	finElink	10
3.2	Business Explorer	11
4	État de l'art	14
4.1	La datavisualisation	14
4.2	Les solutions similaires	15
4.3	Les langages de traitement de données	15
4.4	La diversité des graphiques	16
4.5	Représenter un graphique en Javascript	17
4.6	Complexité du parcours des données	18
5	Déroulement du stage	20
5.1	Langages utilisés	20
5.2	Environnement de travail	21
5.3	Gestion du projet	22
5.4	Méthodes de travail	22
5.5	Réalisations	24
5.6	Améliorations	29
5.7	Évolution du projet	29
6	Bilan	30
A	Exemples visuels de Business Explorer	32
	Bibliographie	34

CHAPITRE 1

INTRODUCTION

Durant l'année de Master 1 MIAGE¹, un stage de trois mois minimum est requis.

Suite à une recherche peu fructueuse, j'ai contacté François Delbot, maître de conférences à l'Université de Paris-Nanterre pour lui demander s'il n'avait pas des offres de stage disponibles. C'est ainsi que j'ai pris contact avec Gaétan Le Chat, mon tuteur de stage et que j'ai pu rejoindre FRS Consulting.

Ce rapport présente le travail effectué du 9 avril 2018 au 20 juin 2018 (remise du rapport). Cependant, celui-ci continuera jusqu'au 31 juillet 2018.

L'objectif du stage est de concevoir et d'intégrer aux solutions existantes de l'entreprise une solution d'analyse, de curation et de valorisation de données stratégiques générées directement ou indirectement.

1. Méthodes Informatiques Appliquées à la Gestion d'Entreprise

CHAPITRE 2

FRS CONSULTING

FRS Consulting est spécialisé en stratégie et financement de l'innovation, des mesures fiscales nationales aux subventions européennes. Le cabinet, fondé en 2011 par 3 experts reconnus de l'innovation, notamment en fiscalité de la recherche, accompagne les entreprises innovantes des biotechnologies et sciences de la vie, du digital, ainsi que les grands comptes industriels en France métropolitaine et d'Outre-Mer, en Belgique, Espagne et au Royaume-Uni. L'objectif de FRS est de booster la croissance des entreprises innovantes à fort potentiel, et ainsi de construire les leaders de demain.

FRS porte l'innovation dans son ADN. Tout son savoir-faire exprime cette conviction. Il est basé tout d'abord, sur une analyse rigoureuse des facteurs scientifiques, économiques et concurrentiels du projet d'innovation. Ce diagnostic permet ensuite d'établir une stratégie de financement public, à moyen et long terme, qui identifie les dispositifs de soutien à l'innovation, nationaux et européens, les plus adaptés au fur et à mesure du développement du projet. Cette compétence s'adresse aussi bien à des projets collaboratifs réunissant plusieurs partenaires qu'à des projets menés par une entreprise seule. Pour chaque projet accompagné, FRS assure un engagement complet et entier, avec une latitude à 360°.

L'humain - ses qualités, ses aspirations, la précision de sa réflexion - est au centre des préoccupations de FRS. Cet engagement assumé est symbolisé par son logo, l'homme en calligraphie chinoise. C'est pourquoi, l'équipe internationale de FRS est réunie autour de valeurs fortes - Confiance, Excellence, Audace. Elle est composée de docteurs et d'ingénieurs en sciences issus du monde de la recherche, notamment en biologie médicale, en informatique ou encore en processus industriels, d'experts en intelligence économique, ainsi que de spécialistes en levées de fonds publics, y compris d'experts-évaluateurs pour la Commission européenne.

FRS dispose également d'un écosystème riche favorisant les échanges entre les principaux acteurs européens de l'innovation : représentants institutionnels nationaux et européens, pôles de compétitivité (Cap Digital, Medicen, Finance Innovation, Alsace Biovalley, Systematic, etc.), laboratoires académiques (Institut de Cardio-métabolisme et Nutrition - ICAN, Centre d'Immunologie de Marseille- Luminy - CIML, INSERM, Laboratoire d'Informatique de Paris 6 - LIP6, Laboratoire EconomiX, etc.), think tanks, ainsi que des partenaires spécialisés. Fort de ce réseau, FRS est sollicité pour contribuer au débat sur l'innovation et la transition numérique, à Paris et Bruxelles. Ses analyses sont également régulièrement publiées dans la presse généraliste et spécialisée. En parallèle, FRS développe son propre projet de recherche pour améliorer les connaissances dans l'évaluation des politiques publiques européennes de soutien à la recherche et à l'innovation.

FRS est convaincu du rôle économique et social que l’entrepreneuriat joue pour insuffler le dynamisme nécessaire à la croissance, développer de nouvelles pratiques et enrichir le tissu économique. Aussi, FRS s’engage depuis plusieurs années, auprès d’associations comme Réseau Entreprendre, ou de pépinières et hôtels d’entreprises afin de transmettre son savoir et son expérience auprès des jeunes créateurs de start-up. Ces échanges marqués de la forte implication de FRS démontrent régulièrement leur efficacité pour faire grandir des projets innovants à répercussions sociétales et environnementales.

L’entreprise est capable d’accompagner ses clients de l’identification des projets à la perception des fonds. Pour cela, elle fait appel à des consultants et rédacteurs scientifiques : deux profils complémentaires assurant la pluridisciplinarité de FRS.

Les rédacteurs possèdent une expertise dans leur domaine respectif et sont amenés à la mobiliser pour gérer les projets auxquels ils sont rattachés. Leur rôle est d’analyser, challenger, et renforcer le processus scientifique envisagé lors du projet et à terme de rédiger et de défendre le projet auprès des organismes financeurs.

Les consultants accompagnent les entreprises en collaboration avec les rédacteurs, néanmoins là où le rédacteur n’intervient que sur des aspects techniques le consultant lui porte les jalons orientés business, il conduit le projet de l’élaboration jusqu’à sa toute fin et intervient surtout pour structurer la stratégie d’innovation afin qu’elle soit cohérente avec les objectifs de court, moyen et long-terme de l’entreprise.

C’est dans ce cadre que s’inscrit mon stage au sein du pôle R&D de FRS. Ce pôle est constitué de :

- Gaétan Le Chat, responsable du pôle,
- Kymble Christophe et Vincent Grollemund, 2 étudiants en thèse Cifre¹,
- Cheikhouna Diouck et moi, stagiaires niveau Master 1.

L’équipe travaille en commun sur un outil, appelé *finElink*, que je détaillerai ultérieurement. Cependant, chacun a une mission précise, ce qui fait que nous travaillons en autonomie sur des fonctionnalités différentes. Cela n’empêche en aucun cas l’entraide entre chaque membre de l’équipe, à la fois par la correction de *bogues* mais aussi par le partage de connaissances ou d’opinions.

1. Conventions Industrielles de Formation par la Recherche

CHAPITRE 3

RECOMMANDATION DE DISPOSITIFS D'AIDE À L'INNOVATION

3.1 finElink

Le projet innovant *finElink* est né du constat de FRS Consulting, partagé par de nombreux acteurs de l'innovation [1], que la multitude de dispositifs d'aide aux entreprises et à l'innovation et d'acteurs publics engagés rendent difficile la recherche et l'identification de financement par les entrepreneurs porteurs de projets innovants. Trois obstacles principaux ont été identifiés : le défaut d'information, le manque d'expertise, le besoin d'un réseau. Ces difficultés ne sont pas insurmontables dans l'absolu, mais elles demandent énormément de temps et de savoir-faire. C'est précisément pour cette raison que certaines entreprises innovantes font appel à des cabinets de conseil qui mutualisent les efforts nécessaires. Aujourd'hui, une solution technique leur permettant de surmonter ces obstacles et accessible à l'ensemble des acteurs de l'innovation est nécessaire pour renforcer la capacité d'innovation en Europe. Malgré une constante progression du nombre d'entreprises innovantes en Europe, 51% des sociétés de plus de 10 salariés dans l'Union Européenne ne sont pas innovantes, et 10% d'entre elles identifient l'accès au financement public comme l'obstacle principal à leur innovation [2]. La réponse conçue et développée par FRS Consulting est la plateforme *finElink*.

finElink est une plateforme applicative d'intermédiation du financement de l'innovation avec pour objectif socio-économique principal de renforcer la capacité d'innovation de la France et de l'Europe. À cette fin, cette plateforme web se propose d'améliorer l'accessibilité et la représentation des informations sur les dispositifs de financement des projets innovants, en évaluant les aides financières sur l'ensemble de leurs externalités (retour technologique, emplois, labellisation pour d'autres financements, etc.) et en recommandant les meilleures sources de financement à chaque projet innovant, et ce gratuitement pour l'utilisateur. Une fois les sources de financement identifiées, les services d'aide à la constitution de consortium et les outils de rédaction des dossiers de demande et de business plan seront proposés par *finElink* afin d'accompagner les utilisateurs tout au long de leurs démarches de financement de l'innovation.

À terme, *finElink* sera un nexus de l'innovation et permettra à l'ensemble des acteurs sur un même outil de rechercher et se faire conseiller des dispositifs de financement ; de s'informer sur la performance des aides financières ; d'identifier les partenaires idéaux pour maximiser les chances de réussite de leurs projets innovants et/ou créatifs ; et de collabo-

rer à la rédaction de leur dossier de demande d'aide. Il introduira également un modèle innovant et disruptif dans le marché du conseil aux entreprises, permettant ainsi à FRS Consulting via ce projet stratégique de se distinguer de sa concurrence.

Le point fort de l'outil est son utilisation d'API¹ REST² mais aussi d'Open Data. Ces technologies permettent d'offrir un service de meilleure qualité en améliorant l'expérience utilisateur.

finElink propose actuellement la possibilité de consulter directement les dispositifs sans passer par de la recommandation. Cela peut être utile à la fois pour les entreprises voulant s'informer des dispositifs actuellement disponibles mais aussi aux personnes internes à FRS qui peuvent alors consulter un annuaire.

On peut donc rapidement distinguer deux types de données de navigations sur *finElink* : la navigation pour la recommandation et celle pour accéder à l'annuaire. À cela s'ajoute des données d'audience du site web générées par Google Analytics³.

3.2 Business Explorer

Suite à la création de *finElink*, FRS récoltait des données de navigation et d'audience or celles-ci n'étaient pas exploitées. Il fallait donc développer une interface rassemblant les informations les plus pertinentes. Ces informations s'adresseront alors à deux "types" d'utilisateurs : en interne chez FRS mais aussi aux financeurs publics.

Pour les financeurs, la plateforme permettrait d'avoir par exemple, un visuel sur l'actualité de l'innovation, mais aussi sur la concordance entre l'argent déployé pour un dispositif et la demande des entreprises innovantes. Cela pourra les aider dans leurs prises de décisions, en accordant plus d'importance ou non à un secteur d'activité, au travail collaboratif ou encore à certaines zones géographiques.

Pour FRS, l'outil permettrait d'aller au plus près de l'information, pour anticiper les secteurs les plus porteurs mais aussi démarcher les entreprises des secteurs émergents. Cela leur apportera une aide à la décision sur la direction stratégique à prendre afin d'identifier les projets les plus porteurs à accompagner.

C'est dans ce contexte que l'outil *Business Explorer* est né, ma mission consistant à mener à bien son développement et son intégration dans l'outil *finElink*.

Comme expliqué précédemment⁴, les données stratégiques de FRS sont générées direc-

1. *Application Programming Interface*, ou Interface de programmation
2. Representational State Transfer, aussi appelé RESTful
3. Service gratuit d'analyse d'audience d'un site web ou d'une application
4. Chapitre 3 *finElink*

tement par la plateforme *finElink* mais aussi indirectement par *Google Analytics*. Il faut alors centraliser toutes ces données pour pouvoir traiter efficacement les informations. Vient alors le besoin de mettre en valeur des informations qui n'étaient pas explicitement présentes dans le système. Par ailleurs, afficher directement toutes les données n'a aucune plus-value, il faut donc convenir des informations pertinentes et les agréger pour pouvoir les exploiter.

C'est pourquoi la première problématique de la mission était de déterminer les informations importantes à récupérer, puis celles à créer à partir de l'existant.

Suite à cela, s'ajoute une problématique de mise en valeur, une difficulté récurrente en informatique. Communément appelée IHM⁵, il est primordial de maximiser le confort utilisateur en fluidifiant la navigation. Une navigation trop compliquée n'est jamais agréable pour l'utilisateur qui risque de ne pas se familiariser avec l'outil et de ne plus s'en servir.

5. Interface Homme Machine

Avant de commencer l'état de l'art suivant, j'aimerais préciser que celui-ci s'est fait avant le début du stage et durant celui-ci. Il n'était pas question de faire un état de l'art uniquement pour le rapport mais plus d'avoir un aperçu du domaine vaste qu'est la datavisualisation. Cela m'a permis d'avoir un regard plus critique sur à la fois mon travail et celui des autres.

Ce chapitre porte donc uniquement sur la recherche et ne donne en aucun cas la réalisation de mon travail à FRS Consulting. Je me suis cependant appuyé sur ces recherches pour le développement de *Business Explorer* mais cette partie sera développée dans un autre chapitre.

4.1 La datavisualisation

La datavisualisation (ou *dataviz*) permet de représenter des données afin d'apporter une aide à la prise de décision. Elle facilite la communication tant en interne qu'en externe d'une entreprise.

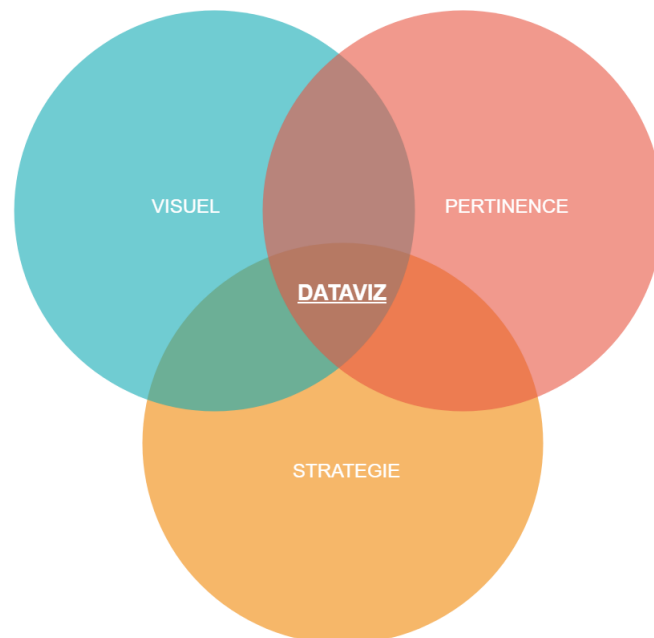


FIGURE 4.1 – Principes de la datavisualisation

C'est un concept qui existe depuis de nombreuses années, bien avant l'arrivée de l'informatique. Elle est cependant très utilisée dans ce domaine au vu du contexte de Big Data qui a émergé ces dernières années.

Le phénomène de datavisualisation repose sur des critères visuels, de pertinence de l'information et de la stratégie qui peut en découler, comme présenté dans la figure 4.1. Ces trois critères sont fondamentaux et permettent de qualifier une solution de datavisualisation.

Le marché de la datavisualisation est encore en pleine croissance au vu de son émergence récente. Néanmoins, il existe déjà des solutions qui répondent aux besoins des entreprises.

4.2 Les solutions similaires

Il existe différentes solutions payantes permettant d'automatiser ou de sous-traiter la datavisualisation des données d'un particulier ou d'une entreprise. Ces outils permettent notamment de lier une multitude de plateformes de données entre elles, comme Google Analytics ou encore Searchmetrics [3].

Ces outils permettent d'améliorer la stratégie marketing la plus utilisée dans le monde professionnel : le SEM¹. Le SEM se divise en deux branches distinctes : le SEO² et le SEA³. Le SEO est une technique de référencement naturel et gratuit. Il consiste à optimiser le contenu des pages web (*on-page*) et la popularité des pages par des liens externes *off-page*. Le SEA ou référencement payant est une technique de référencement passant par la publicité des moteurs de recherche. Elle utilise le principe du PPC⁴ qui consiste à payer l'hébergeur de la publicité selon le nombre de clics obtenus.

Parmi ces solutions, certaines se démarquent de par leur notoriété ou leur innovation. On peut notamment retenir Tableau [4], Dashthis [5], Analytics Canvas [6] ou encore Toucan Toco [7].

4.3 Les langages de traitement de données

«En informatique, une donnée est la représentation d'une information dans un programme» FOUNDATION [8]. Il est donc possible de traiter des données dans le langage de programmation de son choix. Il n'existe pas de réelles contraintes empêchant l'utilisation d'un langage de programmation.

En revanche certains langages et paradigmes de programmation sont plus propices aux traitements de données. C'est pourquoi, on parle généralement de Python, de R ou de

1. Search Engine Marketing
2. Search Engine Optimization
3. Search Engine Advertising
4. Pay Per Click

SAS lorsque le sujet de la datavisualisation est abordé.

- Python [9] est un langage de programmation interprété, orienté objet. Connu pour sa simplicité de lecture, le langage est grandement utilisé par la communauté scientifique tout particulièrement pour les calculs numériques. Par ailleurs, c'est un langage typé dynamiquement, c'est-à-dire que le type de l'objet n'est pas obligatoirement connu à l'avance. Parmi les types les plus représentatifs pour des données, on retrouve les entiers (*int*), les tuples ou n-uplets *tuple*, les tableaux dynamiques (*list*), les tableaux associatifs (*dict*) et les chaînes des caractères (*string*). En plus de la programmation orientée objet, Python permet de programmer selon le paradigme de la programmation impérative. Étant un langage interprété, la traduction du code se fait sans compilateur. Cela permet de faciliter le portage sur n'importe quel système d'exploitation. C'est pourquoi Python est disponible sur n'importe quel système d'exploitation.
- R [10] est un langage de programmation procédurale fréquemment utilisé dans le domaine des statistiques. C'est un langage typé dynamiquement qui permet de faciliter le traitement des volumes massifs de données. De plus, R est aussi un logiciel libre pour les statistiques et la science des données. R (le langage et le logiciel) est *cross-platform*, c'est à dire qu'il est disponible sur n'importe quel système d'exploitation.
- SAS [11] est un langage de programmation ainsi qu'une suite logicielle propriétaire développée par SAS Institute. La suite logicielle utilise le langage SAS pour de l'analyse de données, comme l'analyse prédictive ou l'informatique décisionnelle⁵. SAS est disponible sur les systèmes d'exploitation Windows et UNIX et commence à être porté vers Mac (uniquement une version académique).

4.4 La diversité des graphiques

Un graphique, ou représentation graphique de données, est une représentation visuelle de données chiffrées. Elle s'exprime de différentes manières suivant l'information que l'on souhaite afficher et le message à faire passer. Avant de parler plus en détails des graphiques, il est important de préciser qu'il en existe un trop grand nombre pour en faire une liste exhaustive. De plus, des graphiques peuvent être combinés pour en donner un nouveau.

C'est pourquoi les graphiques présentés ci-dessous ne représentent pas l'ensemble des possibilités mais introduisent les formes régulièrement utilisées dans le domaine de la datavisualisation.

On distingue quatre grands groupes de graphiques :

- les graphiques de relation
- les graphiques de composition
- les graphiques de comparaison
- les graphiques de répartition (ou distribution)

Les graphiques de relation permettent de mettre en valeur les liens entre plusieurs va-

5. *Business Intelligence*

riables. Généralement, ces graphiques se limitent à un petit nombre de variables. Par exemple les nuages de points [12] (*scatter plot*) représentent deux variables tandis que les graphiques à bulles [13] (*bubble plot*) permettent de représenter trois variables plus facilement.

Les graphiques de composition sont utilisés pour comparer une donnée dans un ensemble. Ils représentent généralement les valeurs relatives (pourcentage par rapport au total) mais peuvent également représenter des valeurs absolues. On distingue deux types de graphiques de composition : les statiques et ceux changeant dans le temps. Parmi les graphiques de composition statique, le plus utilisé est le camembert [14] (*pie chart*) car c'est l'un des plus simples à représenter et à comprendre. Pour les graphiques de composition changeant dans le temps, le plus représentatif est le graphique d'aires empilées [15] (*stacked area chart*) qui permet de représenter des valeurs relatives ou absolues.

Pour évaluer l'étendue des données entre elles, on utilise des graphiques de comparaison. Cela permet de trouver rapidement les valeurs les plus extrêmes et celles les plus basses. Ces graphiques permettent aussi de comparer l'évolution des valeurs dans le temps, généralement par augmentation, diminution ou stagnation d'une courbe. Le graphique de barres [16] (*bar chart*) ainsi que le graphique de courbes [17] (*line chart*) sont les plus utilisés parmi les graphiques de comparaison. Attention néanmoins à ne pas confondre un graphique de barres et un histogramme, deux graphiques qui utilisent des barres comme indicateur mais n'ayant pas le même objectif de représentation.

Les graphiques de distribution permettent de savoir comment une variable quantitative est répartie sur un axe défini. L'objectif est de mettre en valeur les caractéristiques particulières d'un jeu de données, comme son intervalle de valeur ou sa tendance générale. Le graphique de distribution le plus utilisé est l'histogramme [18].

4.5 Représenter un graphique en Javascript

JavaScript est un langage de programmation orienté objet à prototype. Il est utilisé dans la plupart des pages web mais peut être utilisé aussi côté serveur. Nous ne parlerons cependant que du Javascript "côté client" car c'est avec celui-ci que les graphiques sont représentés sur internet.

Il existe un grand nombre de bibliothèques JavaScript permettant de faire des graphiques sous deux formats "d'image" : le vectoriel et le matriciel.

Le SVG, ou *Scalable Vector Graphics*⁶, est un format de données utilisé pour créer des ensembles vectoriels sur une page web. Les images vectorielles ne perdent pas de qualité lors d'un redimensionnement puisque ses primitives géométriques⁷ contiennent des attributs sur lesquels des transformations sont possibles (dont la mise à l'échelle).

6. graphique vectoriel adaptable

7. exemple : un segment est une primitive géométrique

Le canvas est un composant de HTML⁸, langage de balisage conçu pour les pages web. Il permet d'afficher des rendus d'image matricielle grâce au javascript. À la différence du SVG, le canvas n'est pas *scalable*, c'est à dire que la qualité diminue lors d'une mise à l'échelle.

Le côté *scalable* du SVG rend ce format plus populaire que le canvas. C'est pourquoi une majorité de bibliothèques JavaScript se sert des SVG comme format de graphique, comme Chartist [19], Plotly [20], D3 [21] ou encore Highchart [22]. Cependant, de grosses bibliothèques utilisent les canvas, comme Chart.js [23] ou CanvasJS [24]. Les bibliothèques sont pour la plupart libres. Parmi les bibliothèques citées, Highchart et CanvasJS sont les seules bibliothèques propriétaires.

4.6 Complexité du parcours des données

	Tableau	Dictionnaire
Copie de la structure	$O(n)$	$O(n)$
Itération sur la structure	$O(n)$	$O(n)$
Récupération d'un élément	$O(1)$	$O(1)$
Mise à jour d'un élément	$O(1)$	$O(1)$
Suppression d'un élément	$O(1)$	$O(n)$
Ajout d'un élément	$O(1)$	$O(1)$

TABLE 4.1 – Complexité des opérations sur les tableaux et dictionnaires Python

La complexité d'un programme correspond à la quantité de ressources requises pour exécuter un algorithme. Elle est généralement représentée par un "Grand O" et permet de déterminer le nombre d'étapes de calcul d'un algorithme. Pour déterminer la complexité du traitement de données, je me suis appuyé sur le document *TimeComplexity* de Python [25]. J'ai choisi ce document car le traitement de données se fera en Python, il était donc pertinent de regarder sa complexité en particulier, bien que celle-ci varie très peu entre les langages. Lorsque l'on prend en compte la complexité d'un algorithme, il faut regarder le pire scénario possible. Cela donne une limite sur le temps maximal que peut prendre un programme.

Parmi les structures de données disponibles en Python, les plus utilisées sont les dictionnaires et les tableaux. Un tableau est un ensemble ordonné d'éléments accessibles par leur indice. Un dictionnaire, ou tableau associatif, est un ensemble de couple clé-valeur dont chaque élément est accessible par sa clé. Il existe différentes opérations suivant la structure de la donnée, en revanche, certaines sont communes et importantes : la copie de structure, l'itération sur la structure, la récupération d'un élément, sa mise à jour, sa suppression et l'ajout d'un élément (à la suite de la liste de données pour un Tableau).

8. HyperText Markup Language

D'après la table 4.1, on peut tout de suite voir que la différence se fait au niveau de la suppression d'un élément. Ce qui veut dire que les deux structures de données ont une complexité similaire. La différence d'accès aux données fait que les tableaux seront plus utilisés pour exploiter des ensembles ou chaque élément n'est pas connu à l'avance. Au contraire, les dictionnaires sont utilisés lorsque l'on connaît la ou les clés et que l'on veut exploiter leurs valeurs.

CHAPITRE 5

DÉROULEMENT DU STAGE

5.1 Langages utilisés

Deux langages ont été utilisés pour le développement de *Business Explorer*, le module sur lequel j'ai travaillé durant mon stage : Python pour le côté serveur (*back-end*) et JavaScript pour le côté client (*front-end*).

5.1.1 Python

Python a été choisi car l'objectif de *Business Explorer* est de s'intégrer facilement à *finElink* dont le développement a été fait en Python. Le fait d'utiliser un autre langage pour une partie d'un site est coûteux en termes de ressources car il faut intégrer le nouveau langage dans l'existant. De plus, cela demandera aux autres membres de l'équipe de développement d'avoir des compétences avancées dans ces deux langages. L'objectif n'est pas de programmer pour soi-même, mais d'avoir un code lisible, compréhensible et facile à entretenir. Par ailleurs Python est un langage couramment utilisé par la communauté scientifique et particulièrement les statistiques. C'est pourquoi dans le cas de *Business Explorer*, il est avantageux d'utiliser ce langage pour manipuler les données. On peut trouver facilement de la documentation ou des réponses pour pallier des problèmes quelconques.

5.1.2 Django

Étant une application web, *finElink* utilise le framework Django pour faciliter son développement. La documentation et les bonnes pratiques du framework en font une technologie réputée et utilisée par de grands noms pour leur site web, comme *Instragram*, *Mozilla* ou encore *Pinterest*. Il utilise un *design pattern*¹ qui lui est propre, le **Model View Template** (MVT). C'est un dérivé d'un patron plus connu qui est le *Model View Controller* (MVC). Le *Model* s'occupe de gérer les données de l'application, le *Template* se charge d'afficher les données à l'utilisateur tandis que la *View* se charge de faire le lien entre le *model* et le *template*. Il faut cependant faire attention aux *View* du MVT et du MVC qui sont deux choses totalement distinctes [26]. La *View* de Django correspond au *Controller* du modèle MVC tandis que le *Template* correspond à un fichier HTML avec du langage de template spécifique à Django. La partie *Model* de Django est constitué d'un *ORM*²

1. patron de conception
2. Object-Relational Mapping

qui fait l'interface entre le programme et la base de données. Cette couche d'abstraction permet de traiter les données de la base comme des objets Python. Django apporte des fonctionnalités de sécurité par défaut contre :

- le Cross Site Scripting (XSS) ³
- le Cross Site Request Forgery (CSRF)
- les injections SQL
- le clickjacking

5.1.3 Javascript

JavaScript [27] a été choisi comme langage *front-end* car c'est le seul employé sur les pages web pour apporter de l'interactivité. Il existe une grande quantité de bibliothèques et de framework.

jQuery [28] a été utilisée pour simplifier le développement des scripts JavaScript. Plus de la moitié des sites web l'utilise, ce qui veut dire qu'il est présent dans la plupart des caches des utilisateurs. De plus, certaines fonctionnalités du design de *finElink* s'appuient sur jQuery, rendant le choix de son utilisation pour *Business Explorer* neutre.

Data-Driven Documents (**D3**) est une bibliothèque de visualisation de données écrites en JavaScript. Elle offre la possibilité de manipuler les SVG, les Canvas et les éléments HTML. C'est l'une des bibliothèques de graphiques les plus utilisées pour les pages web. Les raisons de ce choix sont explicitées à la section 5.5.1.

5.2 Environnement de travail

Le développement de l'application se fait sur un serveur de développement, utilisé uniquement par l'équipe de R&D. Celui-ci contient une base de données similaire à celle du serveur de production. Cependant, il n'est pas pertinent de synchroniser les données car l'évolution de celles-ci ne sont pas suffisamment importantes pour justifier des coûts de migration (manuelle ou automatique). Avoir un serveur de développement distinct d'un serveur de production permet d'avoir un environnement le plus proche de celui de l'utilisateur. Cela réduit la probabilité d'apparition de nouvelles erreurs et du traditionnel "ça marchait sur ma machine pourtant". Un autre avantage du serveur de développement est de ne pas être limité par les capacités de son matériel. Le serveur a des capacités de calcul plus élevées qu'une machine personnelle, ce qui permet de minimiser les ralentissements des opérations sur des grandes structures de données.

Un outil est utilisé pour isoler l'environnement python du projet sur le serveur, il s'agit de *virtualenv*. Celui-ci permet de garder les dépendances requises pour un projet dans un emplacement séparé qui ne concerne que celui-ci. Cela veut dire que deux projets peuvent utiliser une même bibliothèque python dans des versions différentes sans créer des conflits.

3. injection de script dans le navigateur

Pour le développement de l'application, j'ai choisi de travailler "en local". Cela signifie que l'éditeur de texte se trouvait sur ma machine et que le code est envoyé par protocole SFTP au serveur. Cela permet d'avoir l'éditeur de texte de mon choix et de ne pas passer par une connexion SSH pour écrire le code. Cependant, cette connexion SSH était utile pour gérer les fichiers statiques et lancer les tests de l'application.

5.3 Gestion du projet

L'équipe du département R&D s'inspire de la méthode agile pour organiser ses travaux sans toutefois respecter l'intégralité de ses principes. Certains principes sont adaptés au contexte de recherche et développement.

À la place des daily scrum, une réunion est organisée chaque début de semaine. L'objectif est de faire le point sur où en est chacun et ce qui prévu d'être fait dans la semaine.

Le développement de fonctionnalités se faisaient en général sur une à deux semaines, ce qui correspond à des sprints de courte durée. Je me suis impliqué à rendre quelque chose de livrable toutes les semaines de manière à avoir un travail incrémental, régulier et en constante progression.

Étant seul à travailler sur l'outil Business Explorer, organiser des poker planning n'avait pas lieu d'être. Il fallait que j'analyse la difficulté des tâches à faire dans la semaine et que j'en parle avec les membres de l'équipe lors de la réunion hebdomadaire pour trouver ensuite une solution.

Enfin, des cas d'utilisation n'ont pas été écrits pendant le stage car ils avaient déjà été préparés en amont. De plus, *Business Explorer* est un outil principalement destiné aux internes de FRS Consulting, si des doutes subsistaient, je pouvais aller voir directement les utilisateurs finaux de l'outil.

Le projet *finElink* utilise l'outil de gestion de version git et plus particulièrement, la plateforme Gitlab. Chaque membre de l'équipe a une branche pour développer ses fonctionnalités. Ensuite un merge est fait sur une branche de développement pour tester le tout sur le serveur de développement. Une fois que la fonctionnalité est validée, elle est envoyée sur la branche master qui correspond au serveur de production.

5.4 Méthodes de travail

Afin de réaliser l'application Business Explorer, j'ai divisé mon travail en 7 étapes :

- Proposition d'une solution
- Discussion de la proposition

- Développement d'un prototype de la solution
- Validation du prototype
- Amélioration ou changement du prototype
- Test du prototype
- Refactor du code

J'ai décidé de suivre cette méthode car dans un premier temps, elle me permettait d'exprimer mon avis en prenant en compte celui des autres. L'avantage de l'équipe est qu'elle est composée de membres ayant des compétences diverses. J'ai donc profité de mon expérience passée et celle de mes collègues pour enrichir la partie réflexion. Cela m'a permis en particulier, de mieux comprendre comment mettre en valeur des données d'un point de vue financier. Dans un second temps, la spécificité des données de FRS rendait les exemples génériques de datavisualisation peu viables, c'est pourquoi je proposais à chaque fois une première version dite de "prototype" pour avoir la certitude et la preuve de la pertinence du graphique. Une fois celui-ci validé, je rajoutais des améliorations de toutes sortes :

- Mise en place de la sécurisation des données
- Amélioration visuelle
- Ajout d'interaction sur la page

Quand la fonctionnalité était terminée, j'écrivais les tests pour vérifier que le code fonctionnait correctement pour qu'en cas de changement, la stabilité de mon travail ne soit pas impactée. Enfin, du temps était consacré au refactor de mon code pour qu'il soit plus propre, plus facile à entretenir et qu'il respecte les normes PEP8 (*style code*) et PEP257 (*docstring*) de Python. Pour le JavaScript, j'ai suivi les conventions de code ECMA6 car c'est l'une des dernières normes supportées par les navigateurs web.

Lors du refactor de mon code, je me suis inspiré d'une convention⁴ proposée par Mike Bostock [29], créateur de la bibliothèque D3. Cette convention consiste à créer des fonctions génériques pour manipuler des graphiques, de la création à la modification. En plus de cela, l'utilisation du design pattern *method chaining* permet d'exécuter des opérations sur les graphiques de manière plus propre, comme le montre le listing 5.1.

4. *Towards Reusable Charts*


```
function chart(selection) {
  var width = 100; // largeur par défaut
  var height = 100; // hauteur par défaut

  function element() {
    // creation du graphique avec les valeurs par défaut
  }

  element.width = function(value) {
    if (!arguments.length) return width
    width = value;
    return element;
  }
}
// id = identifiant de l'element
var myChart = chart(id).width(500);
myChart.width(); // 500
myChart.width(700); // set la largeur a 700
```

5.5 Réalisations

5.5.1 Comparatif de bibliothèques

La première tâche fut d'établir un comparatif des bibliothèques JavaScript disponibles pour la représentation de graphiques pour ensuite choisir la technologie à utiliser au sein de l'équipe. Je me suis donc penché sur quelques-unes respectant les critères préétablis par mon tuteur que voici :

- La bibliothèque **doit être open source** et la licence **ne doit pas nous obliger à rendre notre outil open source**. *Highcharts* a donc tout de suite été enlevé de la liste car c'est une bibliothèque propriétaire, bien qu'elle soit utilisée par de grands groupes.
- **La taille du(des) fichier(s) de script (et potentiellement de style) ne doit pas être trop importante**, sinon cela ralentit le chargement de la page et impacte l'expérience utilisateur. La notion de taille de fichier est relative à ce qui existe actuellement, par exemple si toutes les bibliothèques ont une taille de 3 Mo sauf une qui est à 30 Ko, celle à 30 Ko sera préférée. Par ailleurs, il semblait pertinent de regarder le temps d'exécution moyen d'une fonctionnalité simple (par exemple un camembert), comme le montre la figure 5.1. Cette comparaison permet d'avoir une autre idée du coût de la bibliothèque côté client.
- Le dernier critère est **d'avoir une unique bibliothèque** assez large pour éviter d'avoir à charger une multitude d'autres bibliothèques spécialisées. Ainsi, l'équipe n'utilisera qu'une seule bibliothèque et ne perdra pas de temps à apprendre l'utilisation des autres.

En plus de ces indicateurs, des critères jugés pertinents ont été rajoutés notamment un

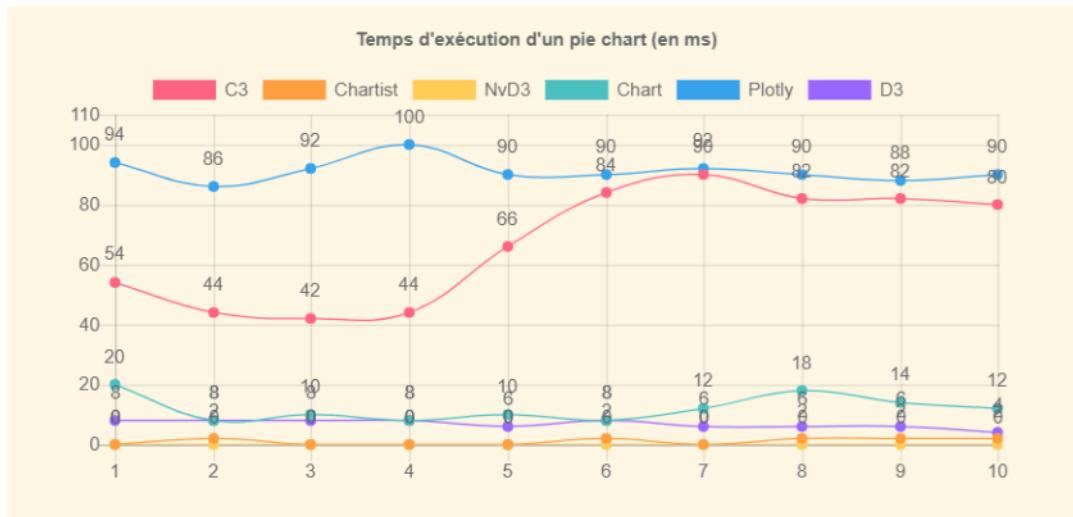


FIGURE 5.1 – Exemple de l'indicateur "Temps d'exécution" des bibliothèques sur 10 exécutions

critère de notoriété de la bibliothèque. Derrière cela l'idée que si un outil est populaire, cela signifie qu'il y a plus de personnes qui l'utilisent et donc une plus grande communauté pour répondre aux questions, corriger des bogues, etc. Par ailleurs les chiffres disponibles sur les dépôts GitHub ont donc été analysés et plus particulièrement le nombre de téléchargelements, de *stars*, de *fork* et de *watch*.

Des critères de maintenance des bibliothèques s'ajoutent ensuite. Il est important de savoir si la bibliothèque répond aux *pull requests* et si les *issues* sont traitées. Dans le cas contraire, si un problème apparaît lors du développement, il y a peu de chance de trouver une solution par ces méthodes. Il est tout aussi important d'avoir une idée de combien de *commit* ont été faits et leur fréquence, pour avoir une idée de la grandeur du projet et s'il est encore entretenu.

La personnalisation qu'offre les bibliothèques - Titre, légende, *tooltip*, *responsive* ou encore zoom du graphique, la possibilité de personnaliser un graphique - est un avantage et évite d'avoir à tout redéfinir soi-même. Cela économise du temps et évite de potentiels conflits avec la bibliothèque comme la redéfinition de fonctions ou le changement d'éléments du DOM.

C'est avec tous ces critères que 6 bibliothèques ont été comparées : C3.js [30], Chartist.js, NVD3, Chart.js, Plotly.js et D3. Parmi elles, C.js3, NVD3 et Plotly.js sont des surcouches de D3 mais n'apportent pas de fonctionnalités de plus que D3, ces bibliothèques permettent juste de produire du code D3 plus rapidement. Par ailleurs, Chartist.js est très limitée en choix de graphiques (3 types : *bar*, *line* et *pie charts*) mais a l'avantage d'être très légère (seulement 40 Ko). Enfin Chart.js est la seule à proposer uniquement des graphiques au format Canvas.

Après concertation, mon choix s'est porté sur D3. Les plus-values des surcouches D3

n'étaient pas suffisamment importantes pour les retenir. *Chartist.js* ne proposait pas suffisamment de solutions pour l'outil à développer, il aurait fallu ajouter une autre bibliothèque. Enfin *Chart.js* était une bonne alternative puisqu'elle proposait des graphiques visuellement intéressants et faciles à construire, néanmoins, le manque d'interaction entre eux et le fait qu'ils soient au format *Canvas* compliquent l'ajout ou la modification de fonctionnalités.

Une fois la bibliothèque choisie, je me suis concentré sur l'utilisation de *Google Analytics*.

5.5.2 Google Analytics

Google Analytics est un service gratuit qui permet de mesurer l'audience d'un site web. On peut accéder à toutes les données calculées depuis une interface web. Dans un premier temps et afin d'éviter une simple duplication de données, une analyse de ce que propose le service était nécessaire. Cela permettait de définir les informations à croiser avec les données de *finElink*. Une notion importante trouvée fut la possibilité de voir la fréquence de visite d'un dispositif de financement de l'innovation pour chaque utilisateur. Cela permet de créer un coefficient de popularité variable selon une deuxième information, comme la répartition des dispositifs dans un secteur d'activité.

Suite à cette analyse, il a fallu apprendre l'utilisation de l'API fournie par Google [31]. Une problématique de nombre de requêtes par jour, limité à 50 000 par jour, est alors apparue. Dans un contexte de développement, et sûrement après la mise en production, ce palier ne sera jamais atteint. En revanche, il semblait nécessaire de prendre en compte le fait de ne jamais atteindre cette limitation. Si pour une raison quelconque, elle était atteinte, l'intégrité du site en serait impactée pour le reste de la journée. Je me suis donc intéressé à l'utilisation d'un cache dans Django.

5.5.3 Cache Django

Django propose 6 caches différents :

- *Memcached*, un serveur de cache basé en mémoire.
- *Database caching* qui stocke les données dans une base de données.
- *Filesystem caching* enregistre le cache dans des fichiers séparés pour chaque valeur.
- *Local-memory caching*, cache par défaut de Django, est une alternative au *Memcached*.
- *Dummy caching* est une implémentation de cache sans action réelle destinée au développement.
- L'utilisation d'un cache personnalisé.

Il était nécessaire d'étudier la performance de chaque cache à l'aide de *benchmarks* et il s'est avéré que le cache par défaut de Django, *local-memory caching* était le plus performant sur les opérations *get*, *set* et *delete* [32]. Il semblait donc préférable d'utiliser ce cache pour pallier le nombre limité de requêtes journalières. La durée de ce cache est modifiable dans un fichier de configuration de Django. J'ai proposé d'implémenter une durée

d'une journée avant de vider celui-ci. Cela permet d'avoir une mise à jour journalière qui peut être particulièrement intéressante en cas de forte utilisation de *finElink*, comme le cas de l'annonce d'un nouveau dispositif de financement. Une durée plus courte n'est pas nécessaire car les prises de décision, de nature stratégique, pouvant découler de *Business Explorer* ne peuvent pas être faites en temps réel.

5.5.4 Visualisation des secteurs d'activité

La première partie de la réalisation graphique de la datavisualisation s'est portée sur les secteurs d'activités des dispositifs de financement de l'innovation. Pour chaque secteur, les données à valoriser étaient :

- la popularité,
- le budget alloué,
- le nombre de dispositifs,
- la répartition géographique (si c'est un dispositif régional, européen, etc.),
- la collaboration est-elle obligatoire ?

Ces informations semblaient être les plus pertinentes à mettre en exergue pour avoir un aperçu des tendances, de l'importance donnée à un secteur d'activité. Par exemple si le secteur de l'environnement dispose du budget de financement le plus conséquent alors que les dispositifs finançant les secteurs informatiques ou de technologies de pointe sont les plus sollicités. Pour éviter l'afflux d'informations, il semblait important d'épurer et donc se recentrer sur l'essentiel, c'est-à-dire la popularité, le budget alloué et le nombre de dispositifs. Cette étape a nécessité une grande réflexion car il fallait identifier les indicateurs les plus importants tout en respectant l'esthétique de la page.

Dans un premier temps, les données de *Google Analytics* sont récupérées, suivies de celles de *finElink*. Ces deux jeux de données (*datasets*) sont ensuite envoyés à une fonction qui s'occupera de les mettre en commun (figure 5.2). Le *dataset* obtenu est ensuite envoyé au template Django pour être affiché.

Une fois le choix des indicateurs fait, le camembert semblait initialement intéressant. Cependant, cela s'est avéré difficile de faire des comparaisons à partir de ce type de graphique. Par la suite, un *sunburst chart* s'avéra plus pertinent pour représenter l'ensemble des informations et pouvoir ainsi mieux les comparer. La répartition de chaque secteur ainsi que l'animation faite lors du changement d'indicateur a permis de représenter dans un graphique chaque secteur, son indicateur et la possibilité de comparer avec l'ensemble des indicateurs (figure 5.3).

Ensuite, pour assurer l'interaction dans le *sunburst chart*, il a été ajouté un tableau de données et un histogramme permettant de positionner une thématique dans son secteur et le secteur dans son ensemble pour affiner les informations qui étaient déjà présentes.

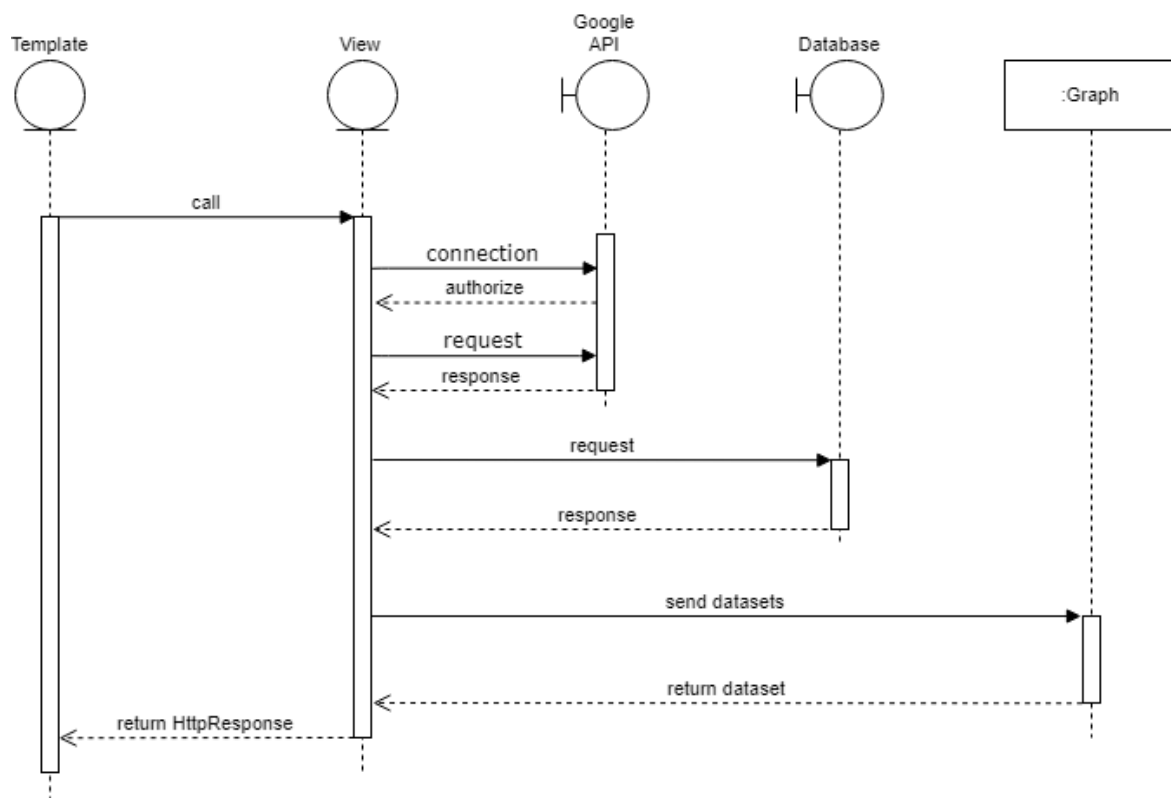


FIGURE 5.2 – Fonctionnement d'un traitement de données de *Business Explorer*

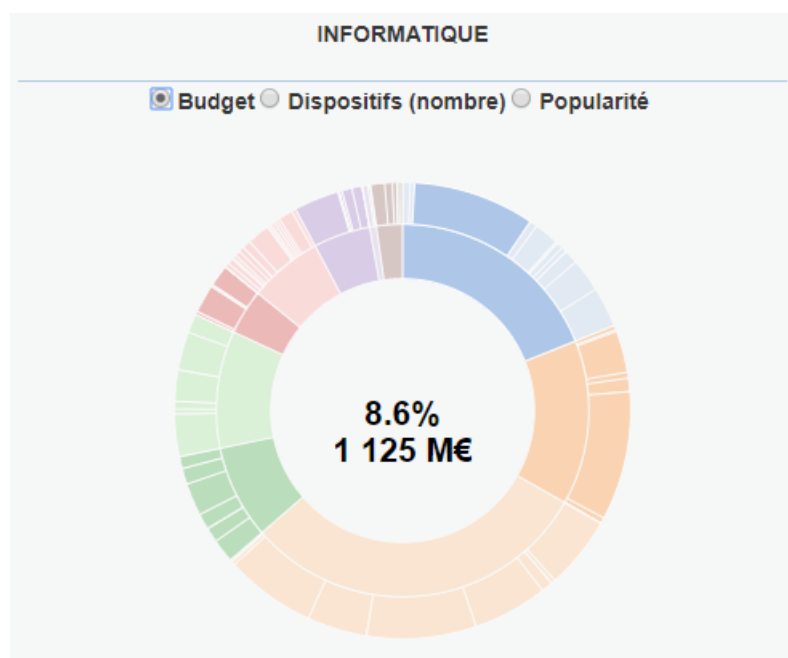


FIGURE 5.3 – Sunburst présentant le budget alloué pour les projets informatiques

5.6 Améliorations

Parmi les améliorations possibles, la première concerne les tests de l'application. Ceux-ci ont été faits pour les parties critiques qui impactaient la validité des données mais il en reste encore à écrire. En particulier la mise en commun des données de Google Analytics avec celles de *finElink*.

La deuxième amélioration serait de pouvoir offrir les informations que j'ai créées dans le temps. C'est-à-dire qu'un modèle de données serait créé pour pouvoir ensuite utiliser ces valeurs et les comparer sur des périodes données (exemple entre 2018 et 2019).

Enfin, des améliorations visuelles aux graphiques peuvent être apportées pour améliorer l'expérience utilisateur. La meilleure solution serait de prendre en compte les retours des utilisateurs finaux en les intégrant à un bêta test.

5.7 Évolution du projet

Parmi les évolutions à venir, la mise en valeur de chaque dispositif unitairement est à concevoir. Celle-ci devrait se faire en un ou deux sprints étant donné qu'il n'y aura pas beaucoup de changements par rapport aux fonctionnalités existantes.

La prochaine fonctionnalité importante est de mettre en valeur les projets récupérés par la recommandation de *finElink*. Ce sera pour moi le moment d'intégrer des forces entre les éléments des graphiques, une première pour moi. C'est aussi une partie très importante pour *finElink* car cela permettra de proposer des solutions de financement personnalisées aux entreprises ou de démarcher de nouveaux clients intéressés par les financements de l'innovation. Il faudra donc s'assurer de l'intégrité des données et soigner l'expérience utilisateur.

CHAPITRE 6

BILAN

C'est avec un bilan positif que se conclut ce rapport de stage. La mission est toujours en cours à la date de remise de ce document et se terminera le 31 juillet.

Les notions apprises durant mon année de Master 1 m'ont permis d'avoir une approche différente des projets habituels. Il m'a fallu faire un réel travail de veille technologique afin de choisir l'outil le plus adapté à ce que je devais développer. De plus, mon apprentissage des *design pattern* m'a permis de réaliser une conception plus générique qui est plus facile à adapter aux besoins par la suite.

Pour les langages et technologies utilisés, mon expérience personnelle en Python ne m'a posée aucun problème de développement. En revanche, il ne m'avait jamais été donné d'exploiter autant de structures de données ce qui est fort appréciable car en plus de gagner en connaissance sur ce point j'ai aussi pu progresser en programmation. Ayant beaucoup utilisé le Javascript pour des projets personnels ou scolaires, ce langage n'était pas nouveau pour moi, en revanche, c'était la première fois que je faisais un projet de cette envergure sans framework tels que React ou Angular. Il en va de même pour les SVG en Javascript, j'ai dû mobiliser une notion clé enseignée cette année : l'abstraction. Elle m'a permis de dépasser les blocages du début et d'assimiler rapidement les principes du SVG.

Certains concepts de la méthode agile ont pu par ailleurs être appliqués. C'est le cas par exemple des sprints. En revanche cette méthode n'a pas été exploitée entièrement car elle n'était pas adaptée au fonctionnement de l'équipe.

ANNEXE A

EXEMPLES VISUELS DE BUSINESS EXPLORER

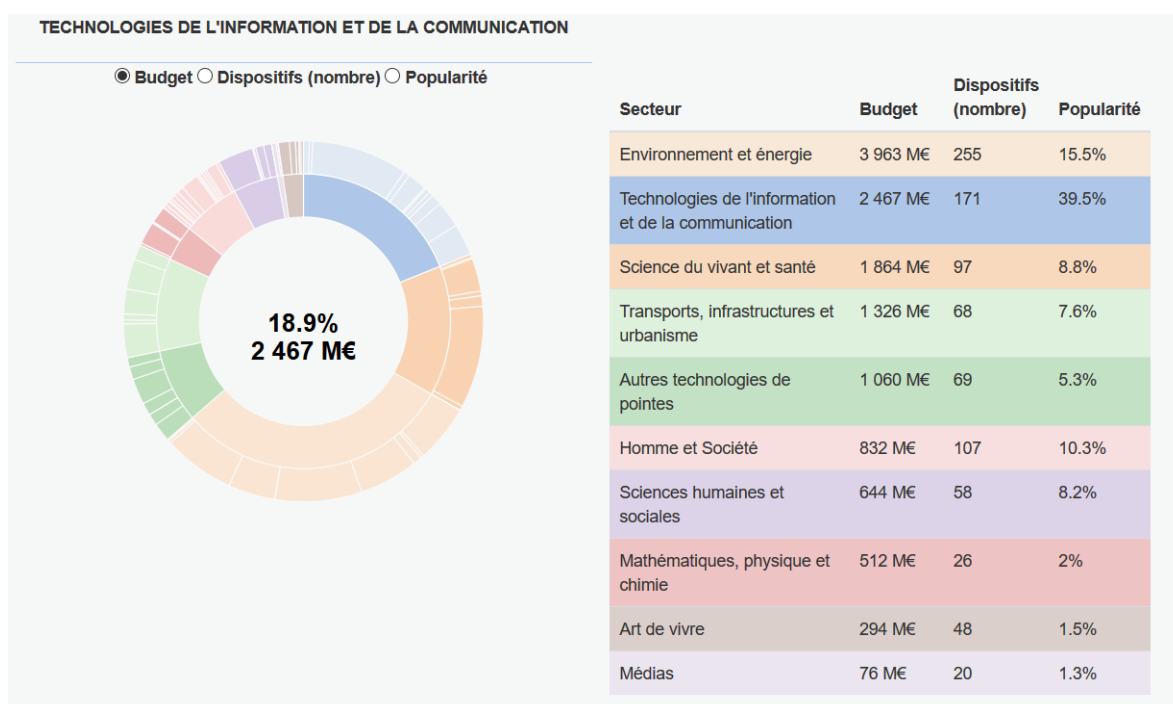


FIGURE A.1 – Vision générale des secteurs d'activité

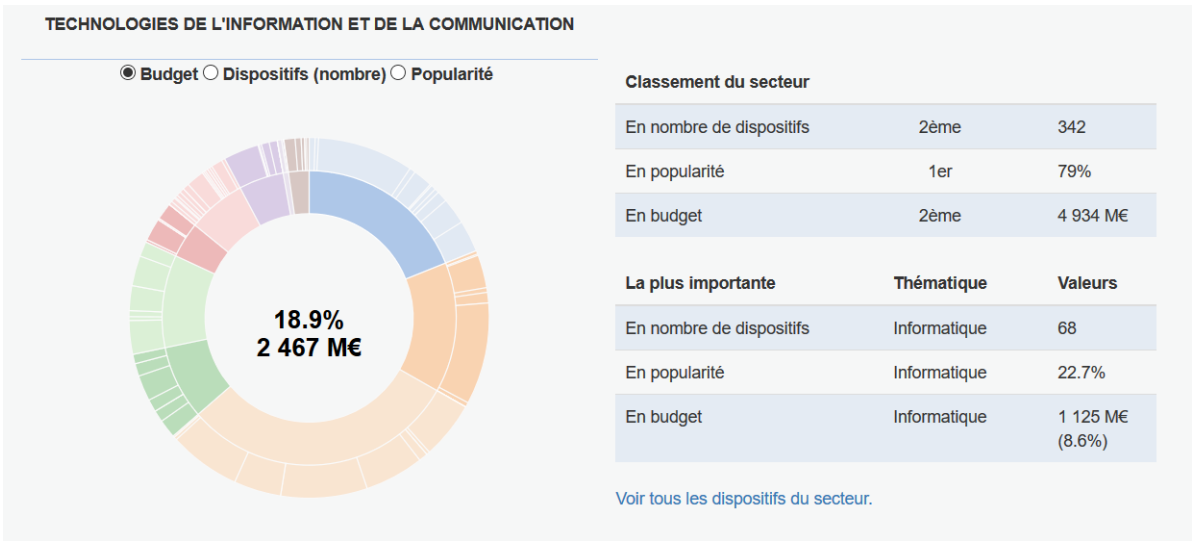


FIGURE A.2 – Vision d'un secteur d'activité

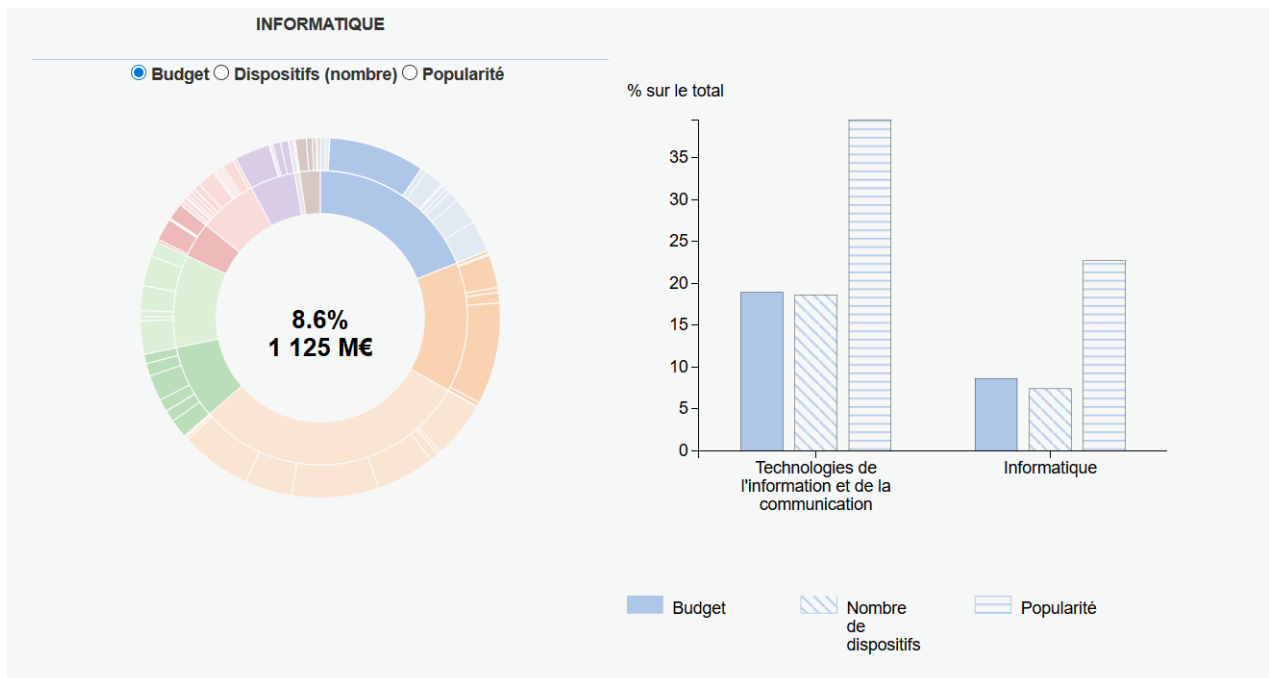


FIGURE A.3 – Vision d'une thématique dans un secteur d'activité

BIBLIOGRAPHIE

- [1] M. Harfi et R. Lallement J. PIZANI-FERY. *Quinze ans de politiques d'innovation en France*. 2016. URL : <http://www.strategie.gouv.fr/publications/quinze-ans-de-politiques-dinnovation-france>.
- [2] European Statistical Data SUPPORT. *Results of the community innovation survey 2014*. 2014. URL : http://ec.europa.eu/eurostat/cache/metadata/fr/inn_cis9_esms.htm.
- [3] SEARCHMETRICS. *Searchmetrics*. URL : <https://www.searchmetrics.com/fr/>.
- [4] Tableau SOFTWARE. *Tableau*. URL : <https://www.tableau.com/>.
- [5] Moment ZERO. *Dashthis*. URL : <https://dashthis.com/>.
- [6] NMODAL. *Analytic Canvas*. URL : <http://analyticscanvas.com/>.
- [7] Toucan TOCO. *Toucan Toco*. URL : <https://toucantoco.com/fr/>.
- [8] Wikimedia FOUNDATION. *Data (computing)*. 2018. URL : [https://en.wikipedia.org/wiki/Data_\(computing\)](https://en.wikipedia.org/wiki/Data_(computing)).
- [9] Python Software FOUNDATION. *Python*. URL : <https://www.python.org/>.
- [10] The R FOUNDATION. *R*. URL : <https://www.r-project.org/>.
- [11] SAS INSTITUTE. *SAS*. URL : https://www.sas.com/en_us/home.html.
- [12] Math is FUN. *Scatter Plots*. 2017. URL : <https://www.mathsisfun.com/data/scatter-xy-plots.html>.
- [13] The Data Visualisation CATALOGUE. *Bubble Chart*. URL : https://datavizcatalogue.com/methods/bubble_chart.html.
- [14] Math is FUN. *Pie Chart*. URL : <https://www.mathsisfun.com/data/pie-charts.html>.
- [15] The Python Graph GALLERY. *Stacked Area Plot*. URL : <https://python-graph-gallery.com/stacked-area-plot/>.
- [16] The Data Visualisation CATALOGUE. *Bar Chart*. URL : https://datavizcatalogue.com/methods/bar_chart.html.
- [17] The Data Visualisation CATALOGUE. *Line Graph*. URL : https://datavizcatalogue.com/methods/line_graph.html.
- [18] The Data Visualisation CATALOGUE. *Histogram*. URL : <https://datavizcatalogue.com/methods/histogram.html>.
- [19] Gion KUNZ. *Chartist.js*. URL : <http://gionkunz.github.io/chartist-js/>.
- [20] PLOTLY. *plotly.js*. URL : <https://plot.ly/javascript/>.
- [21] Mike BOSTOCK. *D3 Data-Driven Documents*. URL : <https://d3js.org/>.
- [22] HIGHCHARTS. *Hicharts*. URL : <https://www.highcharts.com/>.
- [23] CHART.JS. *Chart.js*. URL : <https://www.chartjs.org/>.
- [24] FENOPIX. *canvasJS*. URL : <https://canvasjs.com/>.
- [25] Python Software FOUNDATION. *TimeComplexity*. 2017. URL : <https://wiki.python.org/moin/TimeComplexity>.

- [26] Django Software FOUNDATION. *FAQ : General*. URL : <https://docs.djangoproject.com/en/2.0/faq/general/#django-appears-to-be-a-mvc-framework-but-you-call-the-controller-the-view-and-the-view-the-template-how-come-you-don-t-use-the-standard-names>.
- [27] MOZILLA. *JavaScript*. URL : <https://developer.mozilla.org/fr/docs/Web/JavaScript>.
- [28] The jQuery FOUNDATION. *jQuery*. URL : <https://jquery.com/>.
- [29] Mike BOSTOCK. *Towards Reusable Charts*. URL : <https://bost.ocks.org/mike/chart/>.
- [30] Masayuki TANAKA. *C3.js*. URL : <https://c3js.org/>.
- [31] Google ANALYTICS. *Google Analytics*. URL : <https://developers.google.com/analytics/>.
- [32] Grant JENKS. *DiskCache DjangoCache Benchmarks*. 2016. URL : <http://www.grantjenks.com/docs/diskcache/djangocache-benchmarks.html>.

TABLE DES MATIÈRES

1	Introduction	6
2	FRS Consulting	8
3	Recommandation de dispositifs d'aide à l'innovation	10
3.1	finElink	10
3.2	Business Explorer	11
4	État de l'art	14
4.1	La datavisualisation	14
4.2	Les solutions similaires	15
4.3	Les langages de traitement de données	15
4.4	La diversité des graphiques	16
4.5	Représenter un graphique en Javascript	17
4.6	Complexité du parcours des données	18
5	Déroulement du stage	20
5.1	Langages utilisés	20
5.1.1	Python	20
5.1.2	Django	20
5.1.3	Javascript	21
5.2	Environnement de travail	21
5.3	Gestion du projet	22
5.4	Méthodes de travail	22
5.5	Réalisations	24
5.5.1	Comparatif de bibliothèques	24
5.5.2	Google Analytics	26
5.5.3	Cache Django	26
5.5.4	Visualisation des secteurs d'activité	27
5.6	Améliorations	29
5.7	Évolution du projet	29
6	Bilan	30
A	Exemples visuels de Business Explorer	32
	Bibliographie	34

TABLE DES FIGURES

4.1	Principes de la datavisualisation	14
5.1	Exemple de l'indicateur "Temps d'exécution" des bibliothèques sur 10 exécutions	25
5.2	Fonctionnement d'un traitement de données de <i>Business Explorer</i>	28
5.3	Sunburst présentant le budget alloué pour les projets informatiques	28
A.1	Vision générale des secteurs d'activité	32
A.2	Vision d'un secteur d'activité	33
A.3	Vision d'une thématique dans un secteur d'activité	33

LISTE DES TABLEAUX

4.1	Complexité des opérations sur les tableaux et dictionnaires Python	18
-----	--	----

LISTINGS

5.1 Exemple de *method chaining* proposé par Mike Bostock 24