

2024

Rapport de stage

TELIO ANDRE

SOGAREP



Rapport de stage

Par Telio ANDRE

2^e année Bachelor of Engineering, campus de Tours

Stage du 08 juillet au 30 septembre



Entreprise : Sogarep, 348 Rue de la Musse - 37530 Amboise

Tutrice : Delphine GAUDIN, Responsable contrôle de gestion, comptabilité et data

Sommaire

1- Introduction.....	3
2- Présentation de l'entreprise.....	3
3- Présentation de l'équipe.....	3
4- Introduction du sujet.....	3
5- Avancement du projet.....	4
6- Conclusion.....	12
7- Remerciements.....	13

Introduction

Dans le cadre de ma deuxième année d'études à Supinfo, j'ai effectué un stage de trois mois au sein de la cellule Data Reporting de l'entreprise Sogarep, située à Amboise.

Cette cellule, spécialisée dans le traitement des données, offre une expérience technique et pratique des enjeux auxquels une entreprise comme AXA peut être confrontée. J'ai ainsi pu mettre en pratique mes connaissances, notamment en SQL, tout en découvrant de nouveaux outils pour manipuler et retranscrire les données.

Présentation de l'entreprise

La Sogarep est une filiale d'Axa France fondée en 1979. Spécialisée dans la gestion des frais de santé des contrats d'assurances collectives, elle compte environ 800 salariés répartis entre deux sites : Amboise et Bléré. Cette répartition permet de maintenir les activités même en cas de sinistre sur l'un des sites.

La Sogarep est une entreprise stable, avec plus d'un million de bénéficiaires en santé répartis sur plus de 90 000 contrats.

Présentation de l'équipe

L'équipe que j'ai intégrée se compose de quatre personnes travaillantes en tant que Data Analysts ou Data Scientists : François-Frédéric MARTIN, Yassine ASSAID, Laurent DENIS et Jean GLEDEL. Sous la supervision de Delphine GAUDIN, cette cellule se consacre à divers projets liés aux données ainsi qu'à la création d'outils pour aider l'entreprise et ses collaborateurs.

Les membres de l'équipe utilisent des outils comme WPS pour manipuler les données en langage SAS, PowerBI pour créer et gérer des rapports, et SharePoint pour partager les outils, la documentation et les rendus de données.

Des réunions d'équipe se tenaient hebdomadairement tous les jeudis pour suivre l'avancement des projets. À la suite d'un réaménagement de la cellule en début de septembre, les réunions ont été déplacées au lundi.

Introduction du sujet

Actuellement, la Sogarep utilise un outil de gestion nommé Pégase. Un projet de migration vers un nouvel outil, Open (ou projet River au sein de l'entreprise), est en cours depuis plusieurs mois.

Ma mission consistait à établir une vue portefeuille des entreprises bénéficiaires. Étant donné que la migration est encore en cours, les bénéficiaires sont répartis entre Pégase et Open. Il a été décidé de se concentrer uniquement sur les données présentes dans Open, l'outil Pégase étant en voie de devenir obsolète chez Sogarep.

La vue portefeuille vise à fournir une vision globale du nombre de bénéficiaires, du nombre de contrats en cours, des polices ainsi que diverses métadonnées.

Les entreprises bénéficiaires étant regroupées en pôles, un aspect spécifique aux pôles a été développé pour récupérer d'autres métadonnées, avec pour objectif final la récupération d'informations par entreprise ainsi qu'une historisation des chiffres clés.

Avancement du projet

La première étape de ma mission a été de me former à l'utilisation des outils de la cellule, notamment WPS et le langage SAS. WPS permet de récupérer des jeux de données fournis par le groupe GFP ici et de les manipuler avec le langage SAS. Ce dernier inclut du SQL sans la rigueur de MySQL (il n'y a pas d'obligation de clé primaire par exemple) et permet également une manipulation des données au sens large.

```
* Permet de trier les codes APE par sections;  
DATA SECTIONS;  
  SET WORK.REFERENTIEL_APE;  
  IF Substr(CODE_APE, 1, 7) = 'SECTION' THEN SECTION = Intitules;  
  RETAIN SECTION;  
  IF Substr(CODE_APE, 1, 7) NE 'SECTION' THEN OUTPUT;  
RUN;
```

Manipulation de la table pour trier chaque libellé à la section qui lui est rattaché

Pour la suite, j'ai essayé de me familiariser avec les tables que j'allais utiliser. L'équipe m'a beaucoup aidé en me guidant lorsque je me sentais bloqué. Les jeux de données disponibles ne contenaient pas toujours les colonnes recherchées et pouvaient inclure des contrats ou des polices expirés. Il était donc souvent nécessaire de faire des jointures pour obtenir des tables adaptées, tout en filtrant les données inutiles à la construction du portefeuille.

Les principales problématiques étaient la perte de données et les doublons, qu'il fallait éviter à tout prix. L'objectif était de trouver une source de vérité unique à laquelle se fier.

Chaque table créée devait se traduire en une entité : pôles, bénéficiaires, contrats et polices. Chaque entité étant centrée sur le client (une entreprise = un client).

La première entité sur laquelle je me suis penché a été celle des pôles. Chaque entreprise devait être rattachée à un pôle. J'avais à ma disposition une unité de visibilité pouvant être utilisée comme référentiel pour récupérer le pôle associé.

Un autre aspect pour cette entité a été le code APE, qui sert généralement à identifier la branche d'activité principale de l'entreprise. La difficulté était l'absence de table de correspondance pour ce code APE. J'ai donc utilisé une fonctionnalité de WPS, l'envoi de fichier.

La cellule dispose d'un système de fichier partagé où l'on peut déposer des fichiers accessibles par WPS. Après avoir récupéré une table de correspondance en ligne, je l'ai importée dans WPS et l'ai filtrée pour la rendre exploitable.

```

* Permet de recuperer la table de correspondance entre CODE_APE et Intitule;
PROC IMPORT DATAFILE='~/NAS/S/14. Espace partagé/Les autres répertoires/referentiel_ape.xlsx'
  OUT=liste_APE
  DBMS=xlsx
  REPLACE;
  SHEET='listeCodeApe';
RUN;

```

Import de la table dans WPS

Une fois cette étape terminée, il était important de vérifier l'absence d'erreurs, notamment en s'assurant qu'une entreprise n'était rattachée qu'à un seul pôle, sinon elle était marquée comme Hors Pôles.

Pour vérifier la présence de doublons, j'ai utilisé des macros-programmes, tous similaires entre eux.

WPS permet la création de macro-variables et de macro-programmes. Une fois rédigés et intégrés dans WPS, ces macros peuvent être utilisées ailleurs. Pour maintenir la clarté et la lisibilité de ma mission, j'ai opté pour la création de macro-programmes dédiés à chaque vérification de doublons, évitant ainsi une surcharge sur une seule page.

```

* Verification des doublons;
%macro check_doublons_poles();
  PROC SQL;
    CREATE TABLE CLIENT_NB_POLES AS
    SELECT
      RAISON_SOCIALE,
      Count(DISTINCT POLE) AS NB_POLES,
      Count(DISTINCT ID_ENT_COL) AS NB_IDS
    FROM
      CLIENT_POLE
    GROUP BY
      RAISON_SOCIALE;
  QUIT;

  DATA DOUBLONS;
    SET
      CLIENT_NB_POLES;
    IF
      NB_POLES > 1
      AND
      NB_IDS = 1
    THEN OUTPUT;
  RUN;
%mend;

```

Macro-programme de vérification de doublons pour l'entité pôles

Dans cet exemple, on compte dans un premier temps pour chaque entreprise le nombre de fois où elle apparaît grâce à son ID unique et le nombre de fois où un pôle lui est associé. Puis dans un second temps, on effectue une vérification si deux pôles différents sont présents pour une même raison sociale.

L'entité suivante concernait les polices, avec des métadonnées importantes comme les risques couverts, le pôle et la portabilité. La portabilité s'applique lorsqu'un assuré perd ou quitte son emploi, mais reste assuré pour une certaine période.

La troisième entité concernait les contrats, avec la date de début, le numéro de police associé, le risque couvert et le pôle. J'ai rencontré des doublons inexpliqués dans cette entité. Après avoir discuté avec Yassine ASSAID, j'ai découvert que certains clients étaient des filiales d'autres clients, ce qui expliquait certains doublons dans les numéros de contrat.

La dernière entité a été celle des bénéficiaires, pour laquelle il fallait récupérer les informations telles que nom, prénom, date de naissance et genre. J'ai également ajouté l'adresse e-mail si disponible, ce qui permettrait de mesurer la propension des e-mails par rapport au nombre total de bénéficiaires. Il était aussi nécessaire de déterminer le type de bénéficiaire, qui peut être l'assuré principal, un conjoint, un enfant ou un autre cas. Il fallait également récupérer la portabilité, qui doit être correcte pour le mois en cours.

J'ai pour cela utilisé des macros-variables, car chaque bénéficiaire pouvait être enregistré dans plusieurs garanties, elles-mêmes sous plusieurs polices, avec ou sans portabilité. Il fallait alors obtenir la police et la portabilité les plus récentes pour chaque garantie.

```
* Prend le premier et le dernier jour du mois actuel pour avoir la portabilite;
%let today = %sysfunc(today());
%let DateDebut = %sysfunc(intnx(month, &today, 0, B), date9.);
%let dateFin = %sysfunc(intnx(month, &today, 0, E), date9.);
%let dateDebMsSql = %sysfunc(putn(%sysfunc(inputn(&DateDebut, date9.)), yymmddn8.));
%let dateFinMsSql = %sysfunc(putn(%sysfunc(inputn(&dateFin, date9.)), yymmddn8.));
%let filEnt = DATE_DEB_LIEN_SALARIAL <= &dateFinMsSql;
%let filSor = DATE_FIN_LIEN_SALARIAL >= &dateDebMsSql;
```

Macro-variables utilisées

L'état Noemie, qui fournit des informations sur la télétransmission de chaque bénéficiaire, devait également être récupéré.

Enfin, il fallait ajouter l'entreprise sous laquelle le bénéficiaire était assuré, ce qui permettait également de récupérer le pôle associé. Là encore, des doublons inexpliqués ont été détectés. Après vérification avec Laurent DENIS sur l'outil Open, nous avons constaté qu'il n'y avait pas d'erreur de ma part et que ces doublons étaient probablement dus à des personnes travaillant à mi-temps.

Une fois toutes les entités terminées, j'ai travaillé sur l'historisation, un aspect de plus en plus demandé dans ce type de projet. Pour cela, j'ai créé des tables intermédiaires permettant de résumer chaque table (nombre d'entreprises, contrats, polices, bénéficiaires, et bénéficiaires par type).

L'ajout du pôle à chaque table a été très pratique, offrant une vue consolidée par pôle. J'ai ainsi pu réaliser une table de résumé précisant la date de la dernière mise à jour du programme. L'historisation nécessitant une mise à jour régulière des données, j'ai élaboré deux macros-programmes pour automatiser ce processus.

```

* Cree la table historique si elle n existe pas;
%macro create_histo();
  %if %sysfunc(exist(shared.historique)) = 0 %then %do;
    PROC SQL;
      CREATE TABLE shared.historique AS
        SELECT
          *
        FROM
          WORK.RESUME_GLOBAL
        WHERE
          1 = 0;
    QUIT;
  %end;
%mend;

* Ajoute les données a la table historique;
%macro update_historique();
  PROC SQL;
    CREATE TABLE temp_table AS
    SELECT *
    FROM WORK.RESUME_GLOBAL
    WHERE DATE_INSERT NOT IN (SELECT DATE_INSERT FROM shared.historique);
  QUIT;

  PROC APPEND
    BASE = shared.historique
    DATA = temp_table
    FORCE;
  RUN;
  PROC SQL;
    DROP TABLE temp_table;
  QUIT;
%mend;

```

Macro-programmes utilisés pour l'historisation

Le premier macro-programme permet de créer la table qui va stocker les données d'historisation si elle n'existe pas déjà.

J'ai décidé de faire ce programme en tant que prévention pour la perte de données. De cette manière, le bon fonctionnement de programme est garanti même en cas de sinistre. Cependant, la nouvelle table, si créée, sera forcément vide.

Le deuxième macro-programme quant à lui, permet d'ajouter les données du jour en utilisant une table temporaire pour copier les données puis les coller à notre table d'historisation. Une vérification de date a été ajoutée pour ne pas se retrouver avec des doublons de date qui pourraient poser des problèmes dans le rapport final.

Une fois les tables finalisées, elles ont été envoyées sur SharePoint, permettant leur récupération dans PowerBI. Chaque table a été exportée sous forme de feuille dans un fichier Excel, déposé dans le système de fichiers partagé et ensuite transféré vers SharePoint.


```

* Permet d'envoyer les tables sur un excel puis sur le sharepoint;
%Let fileOutput = '/mutu/FW/users/METIER/SOGAREP/fichiers/out/DST_Portefeuille_River.xlsx';
x "rm %fileOutput";

PROC EXPORT data=work.client_benef outfile=%fileOutput
dbms=XLSX replace; SHEET = "Portefeuille_client_benef"; RUN;

PROC EXPORT data=work.client_contrat outfile=%fileOutput
dbms=XLSX replace; SHEET = "Portefeuille_client_contrat"; RUN;

PROC EXPORT data=work.client_pole outfile=%fileOutput
dbms=XLSX replace; SHEET = "Portefeuille_client_pole"; RUN;

PROC EXPORT data=work.client_police outfile=%fileOutput
dbms=XLSX replace; SHEET = "Portefeuille_client_police"; RUN;

PROC EXPORT data=work.resume_benef outfile=%fileOutput
dbms=XLSX replace; SHEET = "Portefeuille_resume_benef"; RUN;

PROC EXPORT data=work.resume_benef_client outfile=%fileOutput
dbms=XLSX replace; SHEET = "Portefeuille_resume_benef_client"; RUN;

PROC EXPORT data=shared.historique outfile=%fileOutput
dbms=XLSX replace; SHEET = "Portefeuille_historique"; RUN;

%vps_mac_sp_envoyer_fichier(
  site_SP=      https://axa365.sharepoint.com/sites/Sogarep-Blob,
  chemin_SP=    Datasets dédiés,
  fichier_envoi= DST_Portefeuille_River.xlsx,
  dossier_envoi= /mutu/FW/users/METIER/SOGAREP/fichiers/out,
  conflit=      replace
);

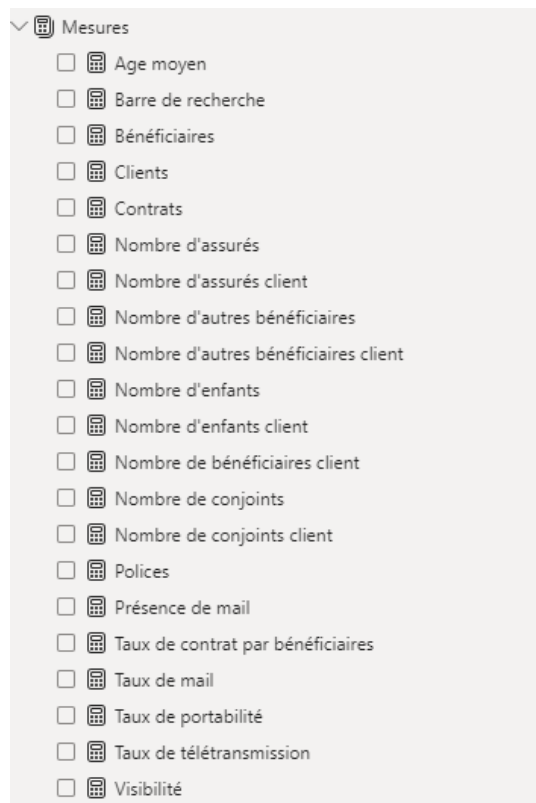
```

Méthode d'export du fichier Excel

La prochaine étape a été la création du rapport PowerBI. J'ai d'abord effectué une formation et réalisé quelques exercices pour me familiariser avec l'outil. Une fois prêt, j'ai commencé le développement du rapport final. PowerBI permet de récupérer directement les tables qui sont sur SharePoint.

Après récupération, certaines données ont dû être transformées, comme le format de date utilisé (yyyymmdd), qui n'est pas reconnu par PowerBI. J'ai donc créé une nouvelle colonne pour formater cette donnée.

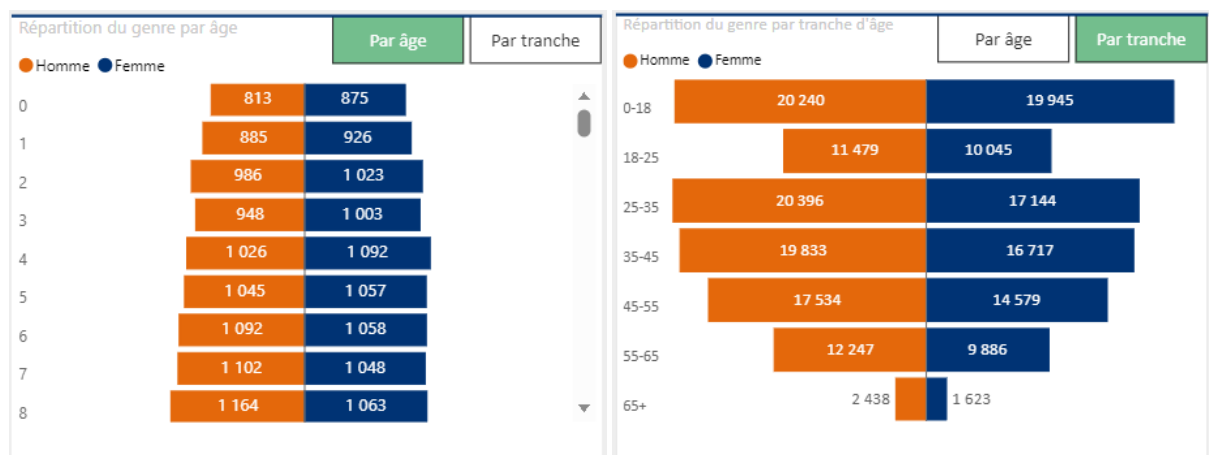
François-Frédéric MARTIN m'a conseillé de regrouper toutes mes mesures dans une même table et de masquer ces colonnes pour simplifier le rapport et éviter de perdre les utilisateurs. Les mesures incluent le nombre de bénéficiaires, de contrats, etc., ainsi que des taux comme le taux de télétransmission, le taux de collecte des e-mails, l'âge moyen et les taux de croissance pour l'historisation.



Liste des mesures

Ces mesures sont faites à l'aide de Power Query. C'est un moteur de transformation et de préparation de données. Il utilise le langage DAX pour cela.

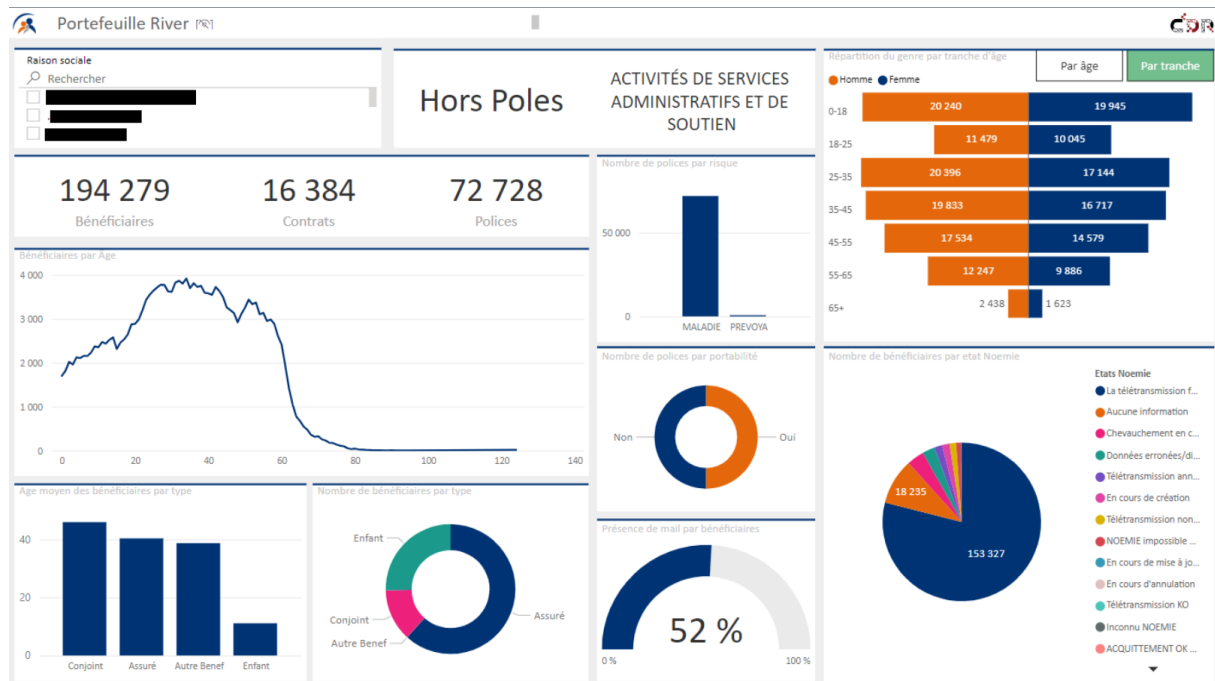
J'ai découvert plusieurs fonctionnalités intéressantes de PowerBI, telles que l'ajout de colonnes conditionnelles, ce qui m'a permis de créer un graphique montrant le nombre de bénéficiaires par tranche d'âge. J'ai utilisé les signets pour sauvegarder des états de page, permettant de visualiser les données sous deux formats différents.



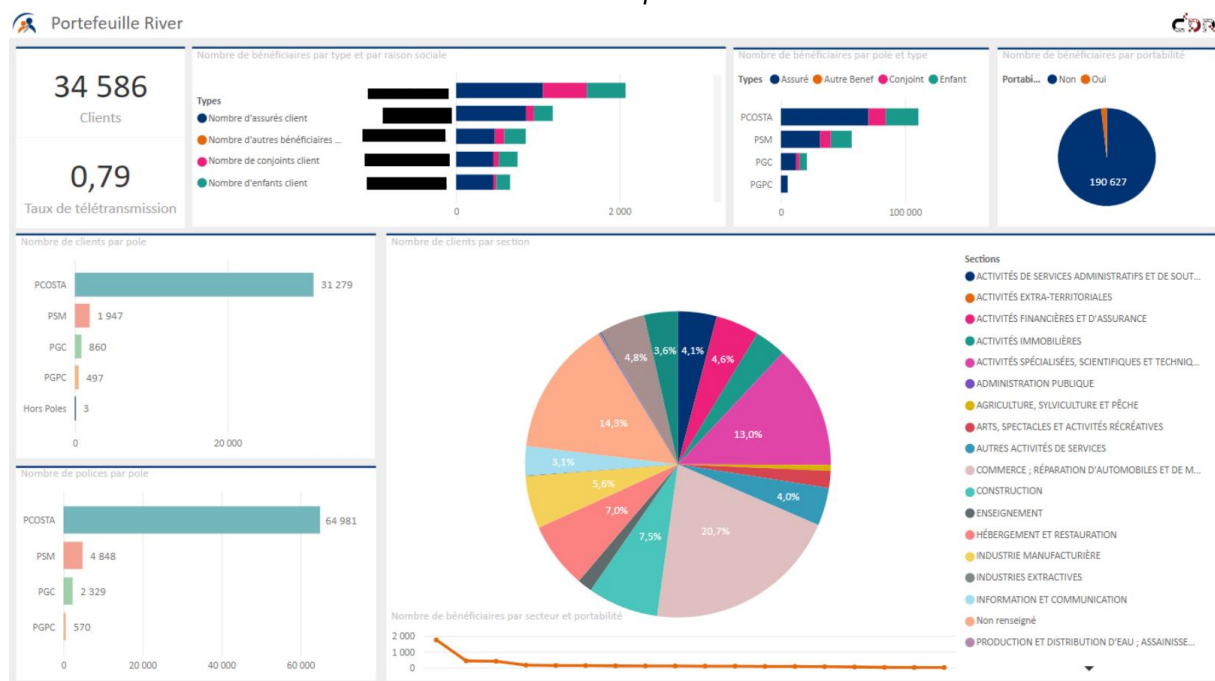
Visualisation par âge et par tranche d'âge

Ce rapport a aussi permis de révéler des erreurs dans certaines tables qui m'ont été données à utiliser. Comme des bénéficiaires avec une date de naissance par défaut (01/01/1900) ou des entreprises de test non nettoyées avec des numéros SIRET identiques.

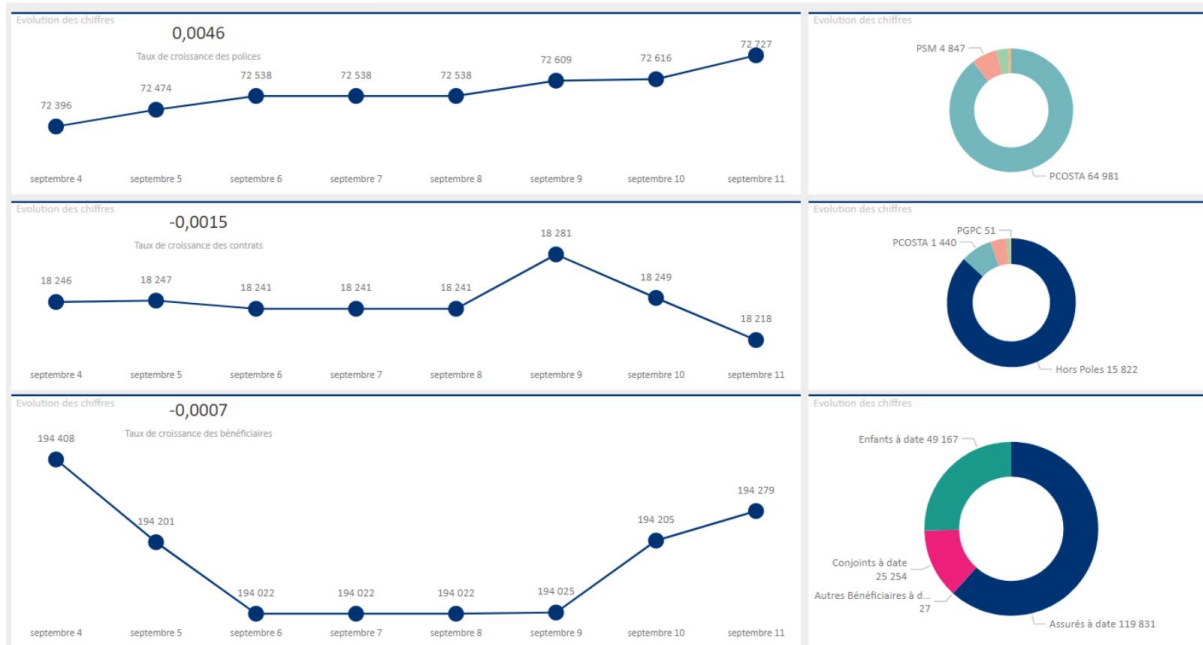
Une fois les différentes pages réalisées, j'ai mis en forme le rapport en me basant sur des rapports existants et en intégrant la charte graphique de l'entreprise.



Vue entreprise



Vue pôle



Historisation

Conclusion

Mon objectif pour ce stage d'été était de travailler dans le domaine de la Data, un domaine qui m'a particulièrement intéressé lors de mes cours. Je ne suis pas déçu de ce choix.

Tout au long de mon stage, j'ai pu mettre en pratique les connaissances acquises au cours des deux dernières années, notamment en SQL et en modélisation de données. Ces compétences m'ont été très utiles pour mener à bien ma mission.

Durant ces trois mois, j'ai découvert les sujets sur lesquels je serai amené à travailler à l'avenir, d'autant plus que mon alternance se déroulera également dans le domaine de la Data. Cette expérience m'a été précieuse, car elle m'a permis d'acquérir des bases solides et de bonnes pratiques que je pourrai utiliser dans mes futures missions.

Enfin, je suis satisfait du travail et du rapport déployé. Je ne pourrais pas voir s'il sera utilisé dans les prochains jours mais j'espère néanmoins qu'il s'avèrera utile pour les personnes l'utilisant.

Remerciements

Je souhaite remercier toute la cellule Data Reporting ainsi qu'à Delphine GAUDIN pour leur accueil et d'avoir accepté pour la première fois un stagiaire dans leur cellule. Toute l'équipe a été présent pour m'aider lorsque j'en avais besoin, malgré les jours de télétravail et les congés de certains, il y avait toujours quelqu'un de l'équipe présent sur site avec moi.

Je souhaite aussi remercier toutes les personnes extérieures de l'équipe avec qui j'ai pu parler pour leur sympathie à mon égard.