

# ORDINAT'R

## Plan de projet



**Picauron Adam**

**Lasserre Joris**

**Olives Matis**

**Descours–Carvel Estève**

**Vlamynck Charles-Adrien**

## **Tables des matières :**

<b>1. CADRAGE DU PROJET</b>	<b>3</b>
1.1 Le projet en une phrase :	3
1.2 Les parties prenantes :	3
1.3 Les Utilisateurs :	3
1.4 Les Délais :	3
1.5 Les Coûts :	3
1.6 Objectif :	4
<b>2. EXPRESSION DU BESOIN</b>	<b>5</b>
2.1 Fonctionnalités principales :	5
2.2 Fonctionnalités secondaires :	5
2.3 Fonctionnalités tertiaires :	5
2.4 Fonctionnalités optionnelles :	5
<b>3. PARTIES PRENANTES, RÔLES ET RESPONSABILITÉS</b>	<b>6</b>
3.1 Les parties prenantes de ce projet sont :	6
<b>4. PRÉSENTATION DE LA SOLUTION</b>	<b>7</b>
4.1 Le PBS	7
4.2 Le WBS	7
<b>5. MANAGEMENT DE L'ÉCHÉANCIER</b>	<b>9</b>
5.1 Planning directeur :	9
5.2 Planning opérationnel :	9
5.3 Planning d'avancement :	10
<b>6. MANAGEMENT DES COÛTS</b>	<b>11</b>
<b>7. LA PROCÉDURE DE RÉCEPTION DES LIVRABLES</b>	<b>11</b>

# **1. CADRAGE DU PROJET**

## **1.1 Le projet en une phrase :**

Réaliser un jeu de données et une interface pour analyser et visualiser les données d'émissions de CO2 et de traitement des déchets d'un pays en Union Européenne.

## **1.2 Les parties prenantes :**

- **MOA** : La SCOP COMPUT'R
- **MOE** : Le groupe d'étudiants ORDINAT'R

## **1.3 Les Utilisateurs :**

Ce sont les employés de l'association COMPUT'R qui vont utiliser le logiciel.

## **1.4 Les Délais :**

La préparation en amont du projet (partie Gestion) a débuté le 06/06/2023 à 8h et se finira le 08/06/2023 à 12h.

Le projet quant à lui va débuter le 08/06/2023 à 14h, il sera terminé pour le 16/06/2023 entre 14 et 17h.

## **1.5 Les Coûts :**

Le groupe ORDINAT'R est composé de 5 membres, l'un d'entre eux (Vlamynck Charles-Adrien) ne sera pas là la première semaine.

Ses coûts se résument donc à 5 jours avec 8 heures / jours (-1h pour le dernier jour car fin à 17h max).

Le reste du groupe ORDINAT'R quand à lui représente :

*Pour la partie gestion :*

3 jours / homme avec 8 heures / jours (-5h car début de la première journée à 9h et fin de la partie gestion à 12h)

Soit 19 heures / homme.

*Pour la partie projet :*

7 jours / homme avec 8 heures / jours (-5h car début de la partie à 14h et fin à 17h max)

Soit 51 heures / homme.

Pour un total de 70 heures / homme.

## **1.6 Objectif :**

Le client est la SCOP COMPUT'R, qui souhaite visualiser les données et les présenter facilement. L'objectif est d'inciter le continent européen à réduire l'émission de gaz à effet de serre en réduisant la production de matériel électronique, et à encourager le recyclage de ces derniers. Le client a besoin d'un moyen de retrouver les données d'un pays relatif à cet objectif afin de pouvoir effectuer des comparaisons.

## **2. EXPRESSION DU BESOIN**

### **2.1 Fonctionnalités principales :**

- Visualiser les données à propos des émissions de CO2 et autres gaz à effet de serre en Europe.
- Visualiser les données à propos de la consommation de produits électroniques pour les pays d'Europe, leurs ventes ainsi que les déchets qu'ils produisent et la façon dont ils sont recyclés.
- Pouvoir comparer les données demandées (à l'aide de graphes ou d'autres outils).

### **2.2 Fonctionnalités secondaires :**

- Visualiser les données à propos des émissions de CO2 et autres gaz à effet de serre en Europe sur les 30 dernières années.
- Des stats permettant de comparer les données obtenues avec d'autres pertinentes (émission par habitants et par km<sup>2</sup> avec la densité de population par exemple).
- Pouvoir afficher la droite de régression entre deux valeurs
- Pouvoir afficher un nuage de points de deux valeurs

### **2.3 Fonctionnalités tertiaires :**

- Trier les pays selon leur émission, consommation de produits informatiques et leur recyclage.
- Donner la possibilité de choisir un intervalle pour faire les graphiques, plutôt que de tout le temps avoir des graphes sur 30 ans.

### **2.4 Fonctionnalités optionnelles :**

- Recommander un pays dans lequel il serait le plus intéressant de mettre en place les activités RRR de l'entreprise.

## 3. PARTIES PRENANTES, RÔLES ET RESPONSABILITÉS

### **3.1 Les parties prenantes de ce projet sont :**

Le client (**MOA**) COMPUT'R :

- C'est de lui que émane l'idée et le besoin du logiciel, il attend un rendu pour le 16 juin 2023 à 17h. C'est lui qui va financer le projet et ne s'y connaît pas assez pour le réaliser lui-même, il détermine la forme du rendu final. Il distribuera le logiciel a ses employés pour qu'ils l'utilisent.

Le groupe (**MOE**) ORDINAT'R :

- Conçoit et développe la solution. Il suit le cahier des charges fourni par le client et a commencé le projet le 6 juin 2023. Il consulte régulièrement le client ainsi que les experts.

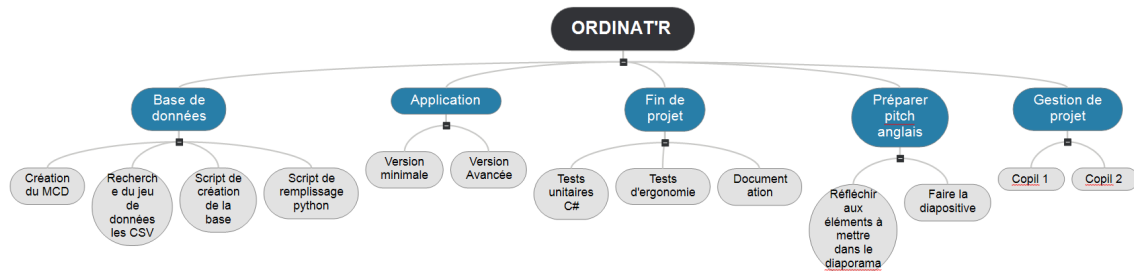
Les employés (**Assistants maîtrise d'ouvrage et experts**) de COMPUT'R :

- Ont connaissance de l'existant, formulent le besoin. Ce sont eux qui vont utiliser le logiciel dans le but de visualiser les données qu'ils souhaitent concernant un pays de L'UE. et pouvoir faire des comparaisons.

## 4. PRÉSENTATION DE LA SOLUTION

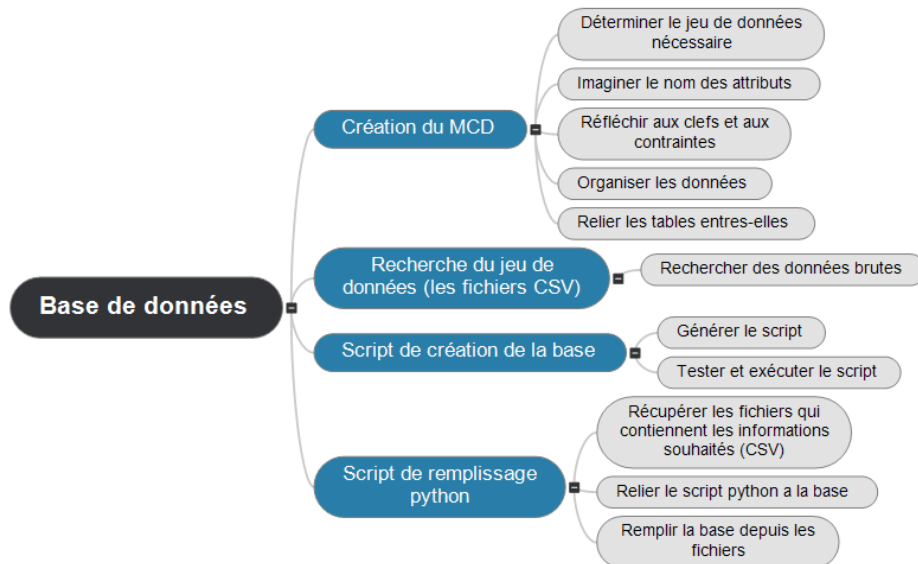
### 4.1 Le PBS

Le PBS permet de visualiser les livrables dans leur globalité :

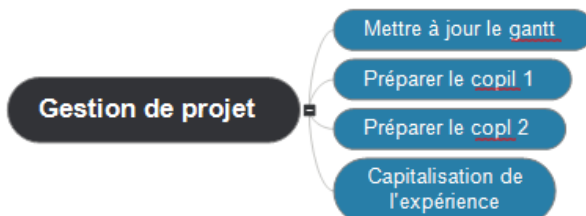
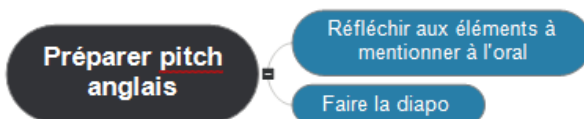
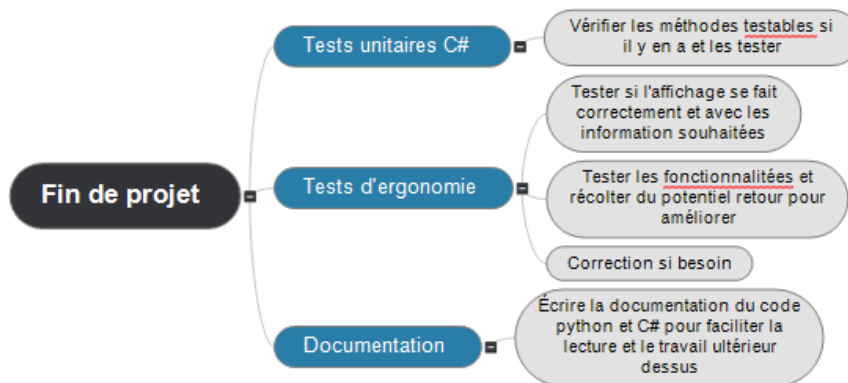
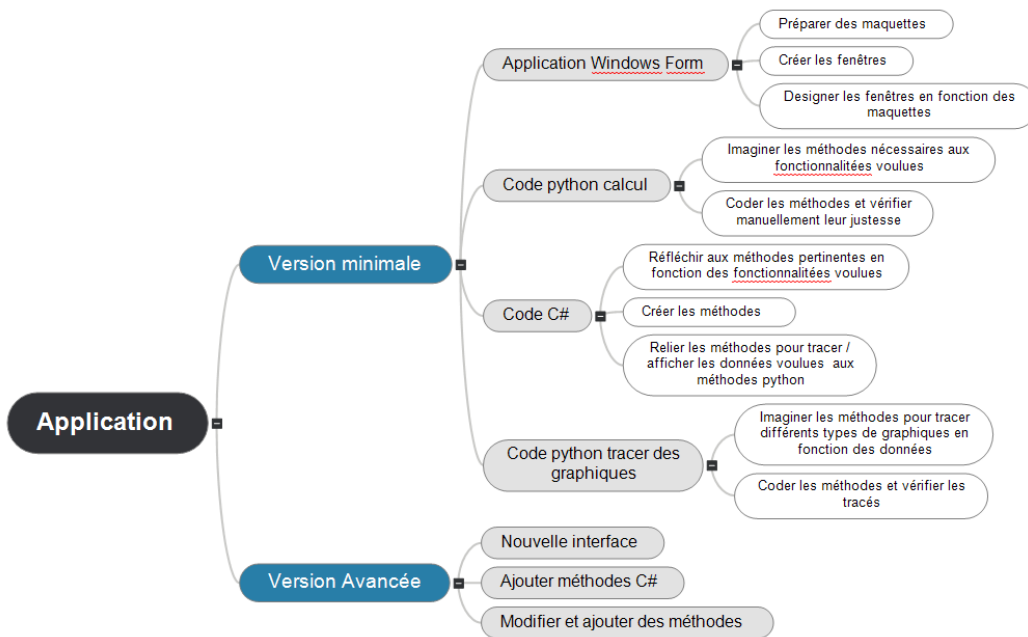


### 4.2 Le WBS

Le WBS permet quand à lui de spécifier et détailler les tâches et les étapes que nous avons trouvés dans le PBS juste avant :



## Projet ORDINAT'R - Plan de projet





[illegible]

	P.	Nom de tâche	Durée	Début	Fin	Préd.	Pro.	Ressource	Totale	
1	-	Project SAE	8 jours	09/06/2023	16/06/23		0%	Joris	211,83 h	
2	-	1. Créer un dépôt git	10 min	09/06/2023	09/06/23		0%	Joris	0,16 h	
3	-	2. Base de données	1,95 j.	09/06/2023	09/06/23		0%	Joris	25,59 h	
4	-	2.1 Création du MCD	0,58 j.	09/06/2023	09/06/23		0%	Joris	5,08 h	
5	-	2.1.1 Déterminer le jeu de donnée.	4 h	09/06/2023	09/06/23		0%	Estève	4 h	
6	-	2.1.2 Imaginer le nom des attributs	10 min	09/06/2023	09/06/23	5	0%	Matis	0,16 h	
7	-	2.1.3 Réfléchir aux clefs et aux co.	10 min	09/06/2023	09/06/23	6	0%	Estève,Matis	0,33 h	
8	-	2.1.4 Organiser les données	15 min	09/06/2023	09/06/23	7	0%	Estève,Matis	0,5 h	
9	-	2.1.5 Relier les tables entre-elles	5 min	09/06/2023	09/06/23	8	0%	Matis	0,08 h	
10	-	2.2 Recherche du jeu de données	0,37 j.	09/06/2023	09/06/23		0%	Estève	3 h	
11	-	2.2.1 Rechercher des données bru.	3 h	09/06/2023	09/06/23	9	0%	Estève	3 h	
12	-	2.3 Script de création	0,31 j.	09/06/2023	09/06/23		0%	Matis	2,5 h	
13	-	2.3.1 Générer le script	1 h	09/06/2023	09/06/23	9	0%	Matis	1 h	
14	-	2.3.2 Tester et exécuter le script	1,5 h	09/06/2023	09/06/23	13	0%	Matis	1,5 h	
15	-	2.4 Script de remplissage	1 jour	09/06/2023	09/06/23		0%	Estève	15 h	
16	-	2.4.1 Récupérer les fichiers qui co.	1 h	09/06/2023	09/06/23	11	0%	Estève	1 h	
17	-	2.4.2 Relier le script python à la ba.	3 h	09/06/2023	09/06/23	16	0%	Estève,Matis	6 h	
18	-	2.4.3 Remplir la base depuis les fc.	0,5 jours	09/06/2023	09/06/23	17	0%	Matis,Estève	8 h	
19	-	3. Application	5,33 j.	09/06/2023	15/06/23		0%	Joris	111,08 h	
20	-	3.1 Version minimale	3,83 j.	09/06/2023	13/06/23		0%	Joris	83,08 h	
21	-	3.1.1 Code python calcul	0,87 j.	09/06/2023	12/06/23		0%	Joris	19 h	
22	-	3.1.1.1 Imaginer les méthodes né.	2 h	09/06/2023	12/06/23	18	0%	Matis,Joris	4 h	
23	-	3.1.1.2 Coder les méthodes et vé.	5 h	12/06/2023	12/06/23	22,18	0%	Matis,Joris	15 h	
24	-	3.1.2 Code python tracer des gr.	0,62 j.	12/06/2023	12/06/23		0%	Estève,Matis	14 h	
25	-	3.1.2.1 Imaginer les méthodes po.	1 h	12/06/2023	12/06/23	22	0%	Estève,Matis	2 h	
26	-	3.1.2.2 Coder les méthodes et vé.	4 h	12/06/2023	12/06/23	25	0%	Estève,Matis	12 h	
27	-	3.1.3 Application Windows Form	1,01 j.	09/06/2023	09/06/23		0%	Joris	16,08 h	
28	-	3.1.3.1 Préparer les maquettes	4 h	09/06/2023	09/06/23	0%	Adam,Joris	8 h		
29	-	3.1.3.2 Créer les fenêtres	5 min	09/06/2023	09/06/23	28	0%	Joris	0,08 h	
30	-	3.1.3.3 Designer les fenêtres en f.	4 h	09/06/2023	09/06/23	29	0%	Adam,Joris	8 h	
31	-	3.1.4 Code C#	2,82 j.	09/06/2023	13/06/23		0%	Joris	34 h	
32	-	3.1.4.1 Réfléchir aux méthodes p.	1 h	09/06/2023	09/06/23	30	0%	Adam,Joris	2 h	
33	-	3.1.4.2 Créer les méthodes	8 h	09/06/2023	12/06/23	32	0%	Adam,Joris	16 h	
34	-	3.1.4.3 Relier les méthodes pour t.	8 h	12/06/2023	13/06/23	33,26	0%	Adam,Joris	16 h	
35	-	3.2 Version Avancée	1,5 jours	13/06/2023	15/06/23		0%	Joris,Adam	26 h	
36	-	3.2.1 Nouvelle interface	2 h	13/06/2023	14/06/23	34	0%	Joris,Adam	4 h	
37	-	3.2.2 Apurer méthodes C#	6 h	14/06/2023	15/06/23	38	0%	Adam,Joris	12 h	
38	-	3.2.3 Modifier et ajouter des métho.	6 h	13/06/2023	14/06/23	34	0%	Matis,Joris	12 h	
39	-	4. Fin de projet	1,12 j.	14/06/2023	15/06/23		0%	Joris	33 h	
40	-	4.1 Tests unitaires C#	0,5 jours	14/06/2023	15/06/23		0%	Joris	8 h	
41	-	4								

### **5.3 Planning d'avancement :**

Le planning d'avancement sera rempli petit à petit chaque jour afin de conserver une trace de l'avancement réel de notre travail. À la fin du projet, nous réaliserons le Gantt final afin de voir les tâches qui nous ont pris plus ou moins de temps à réaliser que prévu, pour que l'on puisse s'améliorer dans nos prochains projets.

## 6. MANAGEMENT DES COÛTS

Coûts prévisionnels :

lot	heures /homme	ORDINAT'R (nb de membres)
Base de données	29	2
Application	100	5
Fin de projet	29	5
Préparer pitch anglais	8	5
Gestion de projet	19	4

Soit environ 819 heures environ de travail au total en prévision pour le projet.

Coûts réels :

Un tableau représentant les coûts réels sera rempli petit à petit au cours du projet afin de garder un œil sur notre progression.

## 7. LA PROCÉDURE DE RÉCEPTION DES LIVRABLES

Les livrables seront restitués sous forme d'un fichier zip contenant le code de l'application et une potentielle documentation sur l'utilisation de l'application ainsi que ses fonctionnalités.

Un document concernant la capitalisation de l'expérience sera produit afin de permettre une mémorisation collective des projets de l'entreprise en construisant une base de données « d'expériences » pouvant profiter à tous.

Un pitch individuel sera également réalisé en anglais afin de parler d'un point en particulier sur le projet.

Tout au long du projet, des "Comités de pilotage" seront présentés sous forme de diaporama afin d'évaluer la progression et consulter les clients / experts.

# ORDINAT'R

## Dossier de spécifications



**Picauron Adam**

**Lasserre Joris**

**Olives Matis**

**Descours–Carvel Estève**

**Vlamynck Charles-Adrien**

## **Table des matières :**

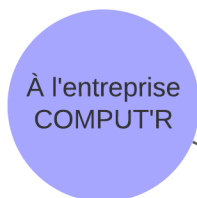
<b>1. Introduction</b>	<b>3</b>
<b>2. Définitions, acronymes et références</b>	<b>4</b>
<b>3. Description générale</b>	<b>5</b>
3.1. Fonctionnalités	5
3.1.1. Fonctionnalités principales	5
3.1.2. Fonctionnalités secondaires	5
3.1.3. Fonctionnalités tertiaires	6
3.1.4. Fonctionnalités optionnelles	6
3.2. Contraintes	6
<b>4. Structure conceptuelle</b>	<b>7</b>
4.1. Modèle conceptuel des Données (MCD)	7
4.2. Schéma de la conception logicielle	8
<b>5. Interface externe</b>	<b>9</b>

# 1. Introduction

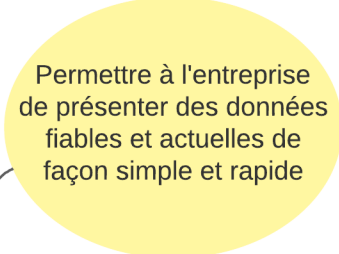
Ce document essaie de décrire les spécifications générales et détaillées de l'application ORDINAT'R. Il explique comment l'application fonctionne, ce qu'elle peut faire pour les utilisateurs mais également les contraintes et les choix de conception des développeurs.

Pour mieux comprendre l'application et ses objectifs, voici un diagramme bête-à-cornes :

**À qui rend-elle service ?**



**Sur qui/quoi agit-elle ?**



**Dans quel but ?**



## 2. Définitions, acronymes et références

Les définitions qui sont originellement données en anglais sont assignées à une traduction.

- **RRR** : Reduce, Reuse, Recycle (Réduire, Réutiliser, Recycler). Définit les actions à réaliser permettant de réduire les déchets, en l'occurrence les déchets électroniques.
- **SDG** : Sustainable Development Goals (Objectifs de développement durable). Ce sont les (17) objectifs tels que décrits par l'ONU afin de garantir le développement durable.
- **Développement durable** : représentation du développement à long terme, incluant des paramètres économiques, écologiques et humains.
- **Gaz à effets de serre (GES)** : gaz participant au réchauffement climatique, dont le CO2 en fait partie.
- **The Shift Data Portal, Eurostat, GISCO** : sites web fournissant un ensemble de jeux de données permettant de recueillir des données capitales pour l'application.
- **Déchets** : un résidu inutilisable, un produit qui a servi.
- **Recyclage** : rendre un produit qui a servi dans le passé pour lui fournir une nouvelle utilité, généralement en récupérant les matières premières de ce dernier.
- **Tendance** : ensemble de données qui évoluent d'une certaine façon à cause de critères spécifiques.

Quelques mots de vocabulaire que nous allons utiliser :

- **MCD (Modèle de Conception des données)** : modèle de représentation des données. Dans ce document, un modèle Entités-associations est présenté, qui permet de décrire les données avec des entités (représentation des acteurs) et des associations (en quoi deux entités sont reliées).
- **Windows Form & C#** : Plateforme de création d'interfaces graphiques. Langage informatique qui, lié à Windows Form, permet de calculer et afficher ce que l'on souhaite.
- **Python** : Langage de programmation, permettant entre autres de remplir une base de données afin de réaliser de la visualisation de avec des bibliothèques intégrées ou de manipuler les données.
- **Framework** : Un Framework contient des composants autonomes qui permettent de faciliter le développement d'un site web ou d'une application en accompagnant le développeur.
- **Booléen** : Type de variable qui représente un état Vrai ou un état Faux (1 ou 0).

## **3. Description générale**

L'application ORDINAT'R permet de visualiser les données relatives aux émissions de GES ainsi que la production, la consommation, les déchets générés par et le recyclage des produits informatiques par pays de l'Union Européenne. Le client pourra donc observer les tendances actuelles des émissions de GES. Cela permettrait de présenter des statistiques intéressantes, et ainsi contribuer au développement durable de façon positive.

### **3.1. Fonctionnalités**

Cette partie décrit les fonctionnalités remplies par l'application, triées par ordre de priorité.

#### **3.1.1. Fonctionnalités principales**

Les fonctionnalités principales sont celles qui suivent :

- Visualiser les données à propos des émissions de CO<sub>2</sub> et autres gaz à effet de serre en Europe.
- Visualiser les données à propos de la consommation de produits électroniques pour les pays d'Europe, leurs ventes ainsi que les déchets qu'ils produisent, la façon dont ils sont recyclés.
- Pouvoir comparer les données demandées (à l'aide de graphiques ou d'autres outils).

#### **3.1.2. Fonctionnalités secondaires**

Les fonctionnalités secondaires sont celles qui suivent :

- Visualiser les données à propos des émissions de CO<sub>2</sub> et autres gaz à effet de serre en Europe sur les 30 dernières années.
- Des statistiques permettant de comparer les données obtenues avec d'autres pertinentes (émission par habitants et par km<sup>2</sup> avec la densité de population, etc ...).
- Pouvoir afficher la droite de régression entre deux valeurs
- Pouvoir afficher un nuage de points de deux valeurs



### **3.1.3. Fonctionnalités tertiaires**

Les fonctionnalités tertiaires sont celles-ci :

- Trier les pays selon leur émission, consommation de produits électroniques et leur recyclage.
- Donner la possibilité de choisir un intervalle pour faire les graphiques, plutôt que de tout le temps avoir des graphes sur 30 ans.

### **3.1.4. Fonctionnalités optionnelles**

Cette fonctionnalité n'est pas demandée par le client et relève de l'ordre du détail. Voici cette fonctionnalité :

- Recommander un pays dans lequel il serait le plus intéressant de mettre en place les activités RRR de l'entreprise.

## **3.2. Contraintes**

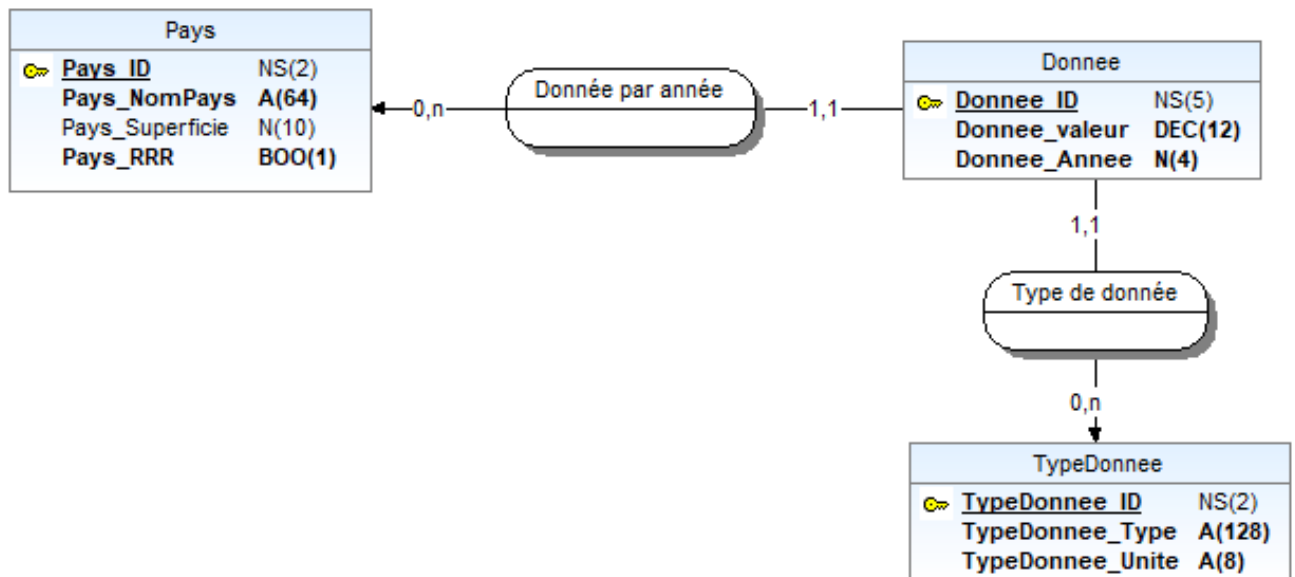
Pour cette application, un groupe de cinq étudiants est impliqué dans sa création.

L'utilisation de Python s'impose, afin de collecter et d'insérer les données dans la base de données servant à gérer les variables désirées. Une autre contrainte logicielle est le framework .NET avec Visual Studio pour la partie interaction application/utilisateur. La base de données doit être hostée sur les serveurs de l'IUT de Gradignan. Enfin, une contrainte importante à prendre en compte est la contrainte de temps, le projet s'étalant sur 6,5 jours + 2,5 jours pour la partie gestion.

## 4. Structure conceptuelle

Plusieurs schémas de conception ont été produits pour ce projet. Voici les schémas en question :

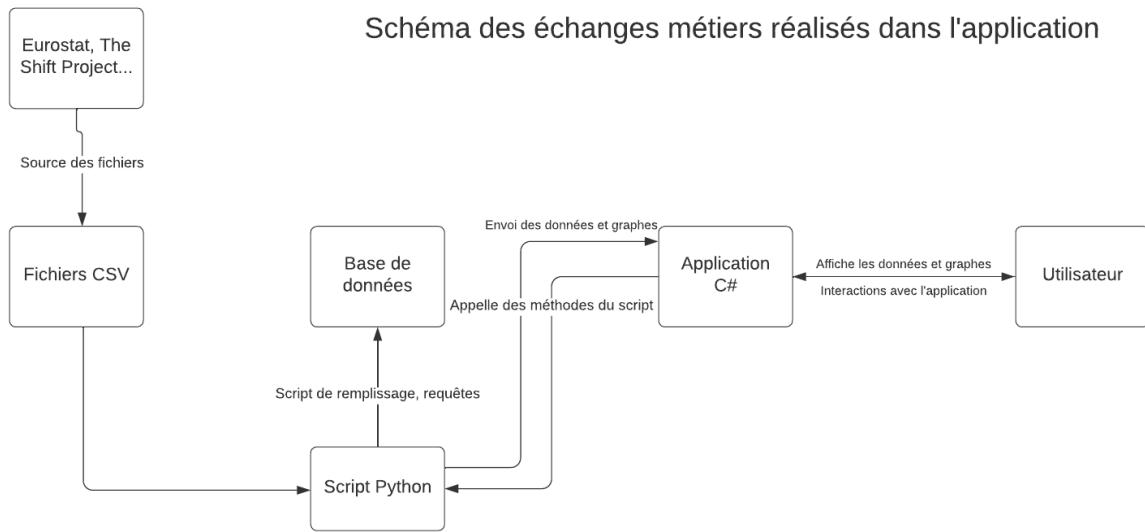
### 4.1. Modèle conceptuel des Données (MCD)



Dans ce modèle de conception des données, un pays est représenté par une entité **Pays**, avec un nom, une superficie, et un booléen indiquant s'il est sélectionné pour les prochaines actions du client.

Une autre entité est l'entité **Donnee**, qui nous permet de récupérer les données d'un pays sur une année, avec la relation **Donnée par année**. Une donnée possède un "type", qui se traduit par sa relation **Type de donnée** avec **TypeDonnee**, qui possède le nom de la donnée et son unité.

## 4.2. Schéma de la conception logicielle



Ce schéma présente les **interactions** entre les parties logicielles de la solution et l'utilisateur.

Tout d'abord, les données brutes sont récoltées à partir des sources que le client spécifie (comme Eurostat ou The Shift Project). Elles sont passées dans un script de remplissage écrit en Python qui permet de remplir la base de données préalablement créée (voir MCD du 4.1).

L'application, écrite en C# avec le framework .NET et l'API Winforms, récupère les données et les graphiques depuis des requêtes du même script Python.

Enfin, l'utilisateur interagit avec l'application, et cette dernière affiche les informations souhaitées.

## 5. Interface externe

Plusieurs maquettes ont été réalisées pour l'interface de l'application. Voici ces maquettes :

Version minimale de l'application :

Listebox

pays 1

pays 2

pays 3

pays 4

pays 5

pays 6

pays 7

pays 8

pays 9

pays 10

pays 3

Emission de Co2 :

Consommation de produits électroniques :

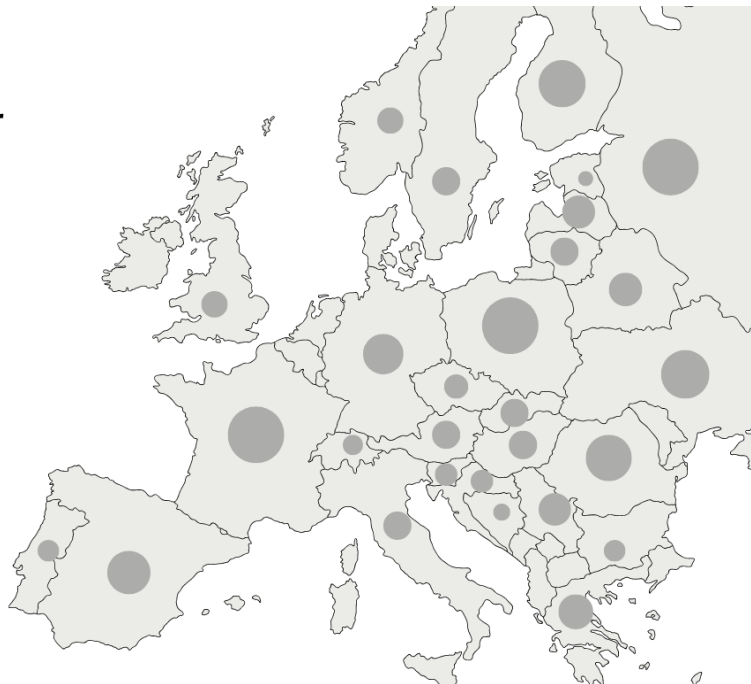
Quantité de Déchets :

Année : 2020

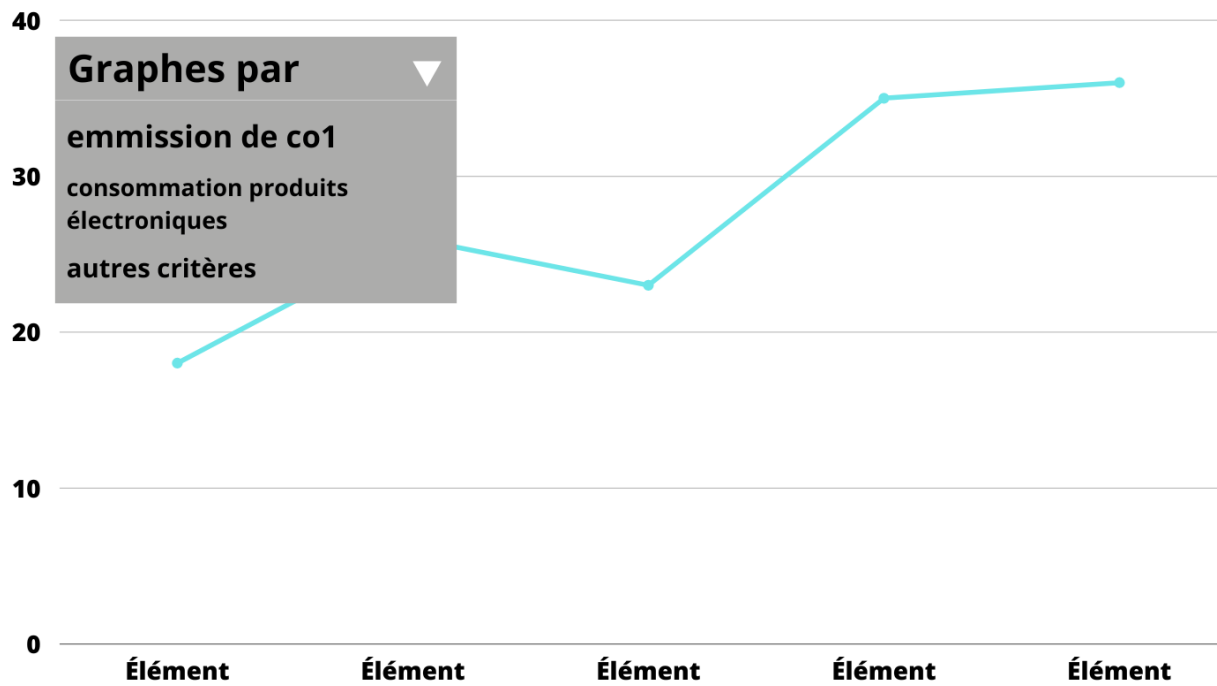
Voir les graphiques

Version avancée de l'application :

Sélectionner sur la carte le pays dont vous voulez avoir les informations



**Pop up qui permet de voir les graphiques/lorsqu'on clique sur les pays (ou sur le bouton dans la version minimale) :**



# ORDINAT'R - Méthodologie utilisée pour la régression linéaire

---

Équipe 7 - Matis Olives, Charles-Adrien Vlamynck, Adam Picauron, Joris Lasserre, Estève Descours-Carvel

---

## Sommaire

1. [Contexte](#)
2. [Problème rencontré](#)
3. [Solution de régression linéaire par morceaux](#)

## Contexte

Pour le projet **ORDINAT'R**, à destination du client **COMPUT'R**, il nous a fallu déterminer des tendances statistiques sur les données désirées. Cela peut se faire à partir d'une régression linéaire, dont le calcul de son équation nous donne le **coefficient de Pearson**. Ce coefficient nous indique si deux données sont en corrélation entre elles si, pour  $r$  le coefficient de Pearson:

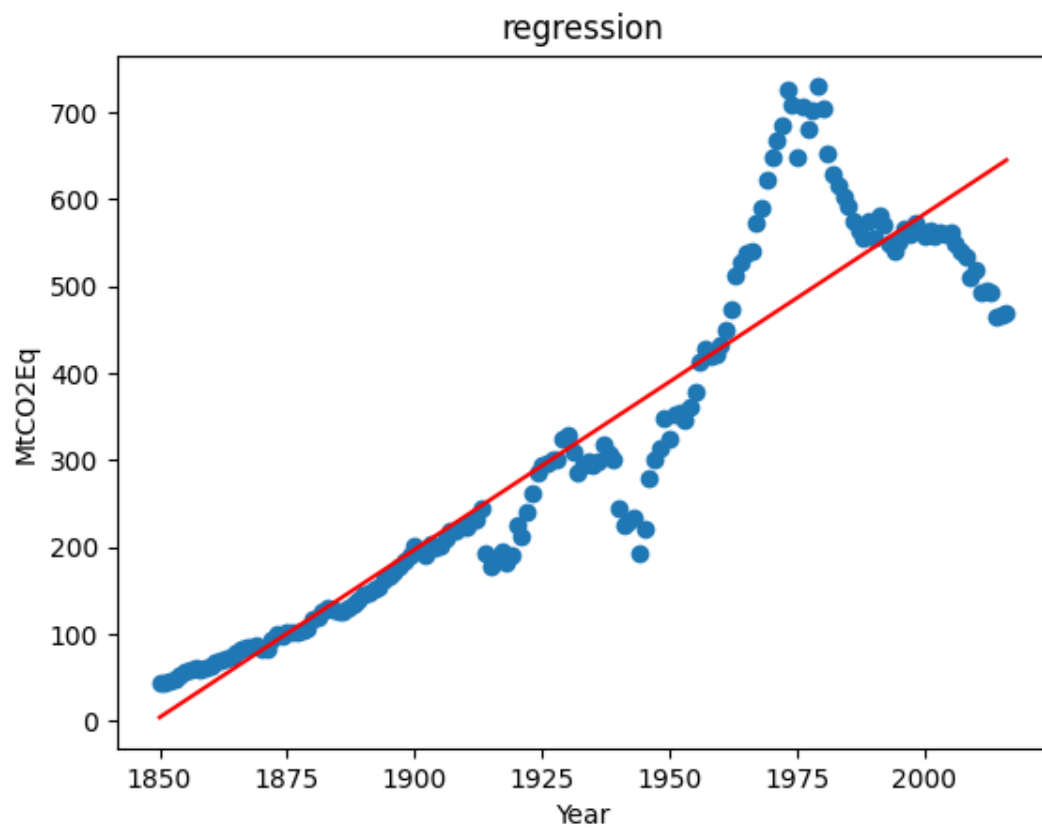
- $r$  s'approche de 1 : les deux variables statistiques sont inversement corrélées. C'est-à-dire que, pour  $X$  et  $Y$  deux variables statistiques, lorsque  $X$  croît,  $Y$  décroît.
- $r$  s'approche de  $-1$  : les deux variables statistiques sont corrélées, et cela signifie que lorsque  $X$  croît,  $Y$  croît également.

Si  $r$  s'éloigne de 1 ou  $-1$ , les variables ne sont pas corrélées.

## Problème rencontré

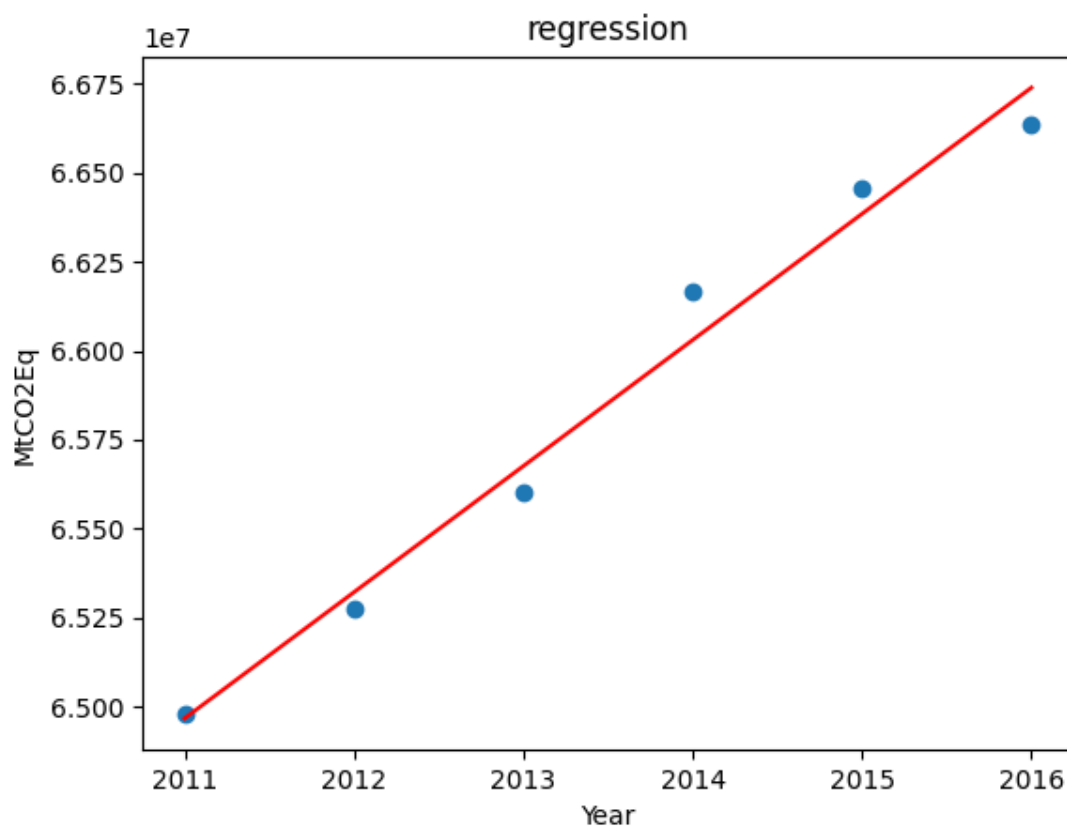
Le problème rencontré lors de l'implémentation de tracé de lignes de régression est que, pour la plupart des jeux de données, elles ne sont pas pertinentes. Soit le coefficient de régression ne se rapproche pas de 1 alors qu'il pourrait sur des intervalles plus petits, soit il se rapproche de 1 sur un jeu trop grand. Dans ce dernier cas, il faudrait découper le jeu en morceaux.

Exemples :



Régression des émissions de gaz à effet de serre et de l'année de la France.

Ici,  $r \approx 0.92$ , ce qui n'est pas suffisant pour déterminer une corrélation ici. Cependant, une conjecture visuelle permet de constater que plus de tendances pourraient être ressorties.



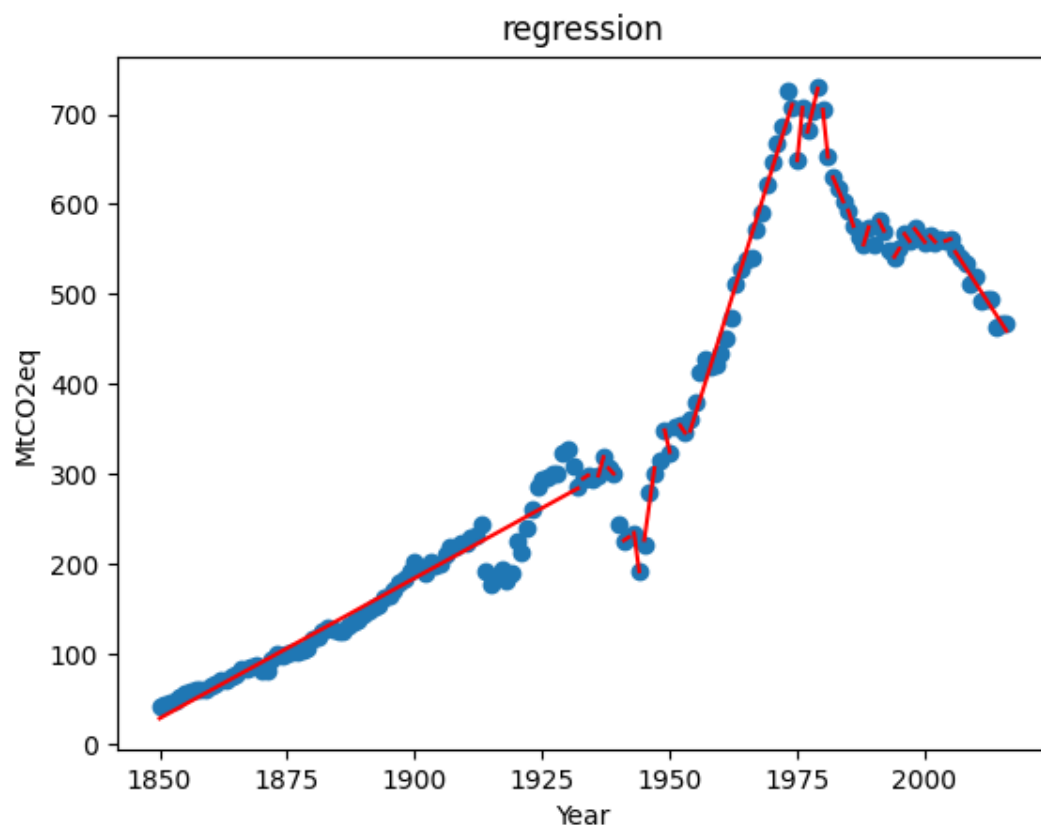
Régression de la population et de l'année de la France.

Ici,  $r \approx 0.99$ , ce qui est suffisant pour ce cas-là car le jeu de données est moins grand.

## Solution de régression linéaire par morceaux

Notre solution pour palier à ce problème est de recourir à un algorithme de régression linéaire par morceaux, avec un algorithme dichotomique. Si le calcul du coefficient de régression  $r$  se rapproche de 1, l'algorithme s'arrête. Sinon, il continue en prenant les deux moitiés du dataframe en entrée, représentant  $Y$ . À la fin de l'algorithme, on se retrouve avec un graphique contenant les lignes de régression pertinentes, pour observer les tendances intéressantes.

Exemples :



Régression par morceaux des émissions de gaz à effet de serre et de l'année de la France.

Le cas reste le même que le précédent car la première itération de l'algorithme trouve  $r \approx 0.92$ . Cependant, dans ce algorithme, si  $-0.95 < r < 0.95$ , le jeu de données est découpé en deux, et l'algorithme est répété jusqu'à obtenir un coefficient satisfaisant.





Régression par morceaux de la population et de l'année de la France.

Le cas est le même que le précédent car  $r \approx 0.98$  à la première itération. Il n'y a donc pas d'autre tendance intéressante à observer dans ce jeu de données. On trouverait peut-être d'autres tendances s'il était plus grand.