université PARIS-SACLAY | FACULTÉ DES SCIENCES D'ORSAY

Bilan du Projet Données Libre

Étudiants: Fabio Alves, Victor Robert-Lambrecht

Groupe: L1 MI4

Professeurs: Thiery Nicolas, et Tamayo-Jiménez Daniel

Université: Paris Saclay

Année: 2020-2021

Projet réalisé: Données Libres

Table des matières:

Répartition du travail:

Afin de répartir le travail équitablement, on a chacun fait un exercice sur deux, respectivement.

Fabio a fait tous les exercices pairs (2,4,6,8), et Victor tous les exercices impairs (1,3,5,7,9 + 10,14) plus l'exercice 10 (puisque les exercices deviennent de plus en plus durs au fur et à mesure qu'on avance dans le projet et que pour chaque exercice que Victor a fait, Fabio en a fait un légèrement plus compliqué), et l'exercice trèfle (14).

Fonctionnalités implantées:

Tous les exercices (de 1 à 9) ont été implantés comme indiqué.

Les exercices 10 et 14 demandaient une certaine autonomie, et leur implantation est donc décrite ci dessous.

Concernant l'exercice 10, la partie subtile était de convertir les coordonnées (longitude et latitude) que le texte csv nous donnait pour chaque commune en coordonnées (x,y). Pour cela, Victor a ouvert google maps, et a déterminé la latitude et la longitude de la France (les valeurs minimales (nord-ouest) et maximales (sud-est)). Ensuite en utilisant des mathématiques linéaires, une fonction (mathématique pas informatique) qui convertissait les coordonnées (longitude, latitude) fut crée, et implantée.

Une fois les coordonnées (x,y) de chaque commune trouvée, les communes furent affichées à l'écran à l'aide de cercles proportionnels à leur nombre d'habitants. La taille de ces cercles peut être modifiée/optimisée avec les flèches, gauche-droite du clavier.

Le barycentre de population fut ensuite calculée, et affiché à l'écran comme un carré bleu.

Finalement, des statistiques sur les données ont été crées et affichées en haut à gauche de la fenêtre.

Pour l'exercice 14, une base de donnée, qui contenait la taille d'une entreprise, et son investissement dans la recherche et développement (d'autres informations étaient présentes mais pas utilisées) a été analysée.

Le but était de voir si il y avait une relation entre la taille d'une entreprise et son investissement dans la recherche, et si on pouvait afficher cette relation (avec un graphe/plot peut être).

Cet exercice a d'abord était fait en Python, Victor étant plus à l'aise avec ce langage de programmation, et aussi parce que Python donnait accès à des bibliothèques d'analyse de données (nommément: matplotlib) qui étaient faciles à utiliser.

La relation entre l'investissement et la taille d'une entreprise était non linéaire: les petites entreprises investissaient plus que les moyennes et les grandes, mais les très grandes entreprises investissaient le plus de toutes les tailles.

Le problème étant qu'on ne connaissait pas exactement la taille d'une entreprise (la taille étant le nombre de personnes y travaillant) mais seulement une tranche de taille: on ne pouvait donc pas calculer l'investissement relatif, mais seulement l'investissement absolu; Il était donc normal que les très grandes entreprises investissaient plus d'argent dans la recherche puisqu'elles ont un plus grand capital monétaire pour commencer.

Aucune véritable relation peut alors être démontrée.

Ces résultats furent envoyés au professeur Nicolas Thiery, qui recommanda de refaire cet exercice en c++.

La bibliothèque graphique sfml a été utilisée pour l'affichage des données.

Les données furent traitées comme pour python, mais l'affichage était plus complexe parce que la fonction plot() a du être programmée à la main, sans bibliothèques. (On aurait pu manuellement dessiner les axes et les données, mais ça n'aurait marché que pour ces données; la raison de créer une fonction plot était de pouvoir visualiser n'importe quelles données).

La fonction plot dessine d'abord les axes, et les valeurs minimales et maximales, aux extrémités des axes (draw_landmarck()), ensuite elle dessine les points de données sur le graphe.

Ensuite afin d'établir une relation, une approximation de la courbe fut crée grâce à l'interpolation lagrangienne. J'aurais préféré utiliser une régression linéaire polynomiale, comme la fonction np.polyfit() le fait en Python, mais je n'ai pas eu le temps de l'implanter.

Finalement la fonction plot() dessine les titres des axes (équivalent en python à xlabel et ylabel).

Difficultés rencontrées

Fabio:

-Utiliser les templates

-La différence de formatage entre les fichiers txt et csv

-utiliser la commande g++ et make au bon endroit

-Le fait de devoir débogguer avec la console (c'est très différent que de la faire avec une IDE)

Victor:

-Savoir quand utiliser la commande g++ et make

-Documenter le code proprement

-Répartir le code en fonctions, à la place de tout mettre dans main

-Savoir si les résultats de l'exercice 9 sont bons (parce qu'il n'y a pas de tests)

Global:

-Peu de grosses difficultés globalement

-On s'est mal organisé, et dans certains exercices on avait besoin du code qui avait été fait dans un exercice précédant: Au lieu de demander au collègue de travail ce qu'il avait fait, on a juste refait le code nous même, ce qui nous a fait perdre du temps. Ce problème a été corrigé par la suite en créant un GitHub, sur lequel on a pu déposer nos fichiers.