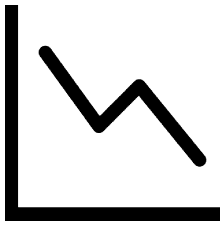
Rapport de projet CY-Météo

Processus et étapes de réalisation



Romain GARDET
Fayçal HACHEM

Groupe6
MI

Table des matières

Respect du cahier des charges

La version fonctionnelle demandée du projet « CY-Météo » se doit de répondre à certains critères. Le but principal de ce projet est de traiter un fichier de données météorologiques pour à terme afficher des graphiques.

Les langages de programmation SHELL et C permettent de répondre à cette problématique. Le SHELL doit permettre à l'utilisateur d'indiquer quelles données il souhaite analyse et filtrer puis afficher le graphique. Le C lui nous permet de trier de façon efficace la masse de donnée très importante du fichier original.

Notre version du projet

Notre version fonctionnelle du projet « CY-Météo » contient toutes les fonctions nécessaires aux instructions. Cependant, selon les options demandées par l'utilisateur le script peut mettre plus ou moins de temps à s'exécuter (max 1min). L'utilisateur peut sélectionner toutes les options à sa guise avec pour seul limite le fait que l'option existe bien.

Le projet à bien été réalisé entièrement à l'aide du Script SHELL et du C.

Sur quels points notre script ne respecte pas le cahier des charges ?

- *Nous avons dans un premier temps rencontrés quelques difficultés sur le C (cf. Les difficultés rencontrées lors du projet). Il n'y a donc pas de lien entre le C et le Shell, toutefois le travail à été réalisé nous avons donc pris le soin de mettre notre code en C disponible sur le répertoire GIT. Le script a cependant été très bien optimisé afin de rendre l'exécution du projet la plus rapide qui soit (max 1min de temps d'exécution).*
- *Le tri des dates en script Shell (option –d) n'est pas fonctionnelle car manque de temps.*
- *Les graphiques vectoriels et avec marge d'erreurs ne sont pas disponibles.*

Contraintes

Nous avons fait face à une grosse contrainte de temps. Cela qui nous a poussés à faire un projet qui n'est pas terminé entièrement. Nous avons fait ce choix afin de privilégier un projet partiellement fonctionnel plutôt que de vouloir tout faire mais que rien ne fonctionne.

Choix des algorithmes selon le besoin

Généralités

Nous sommes conscients que certains compilateurs ne respectent pas les accents, les apostrophes ou les caractères spéciaux. Leur absence est facultative car nous en avons tenu compte lors de l'écriture du programme.

Nous avons réussi à travailler sur les deux langages de programmations différents afin de produire un travail de qualité.

Choix du C

Concernant le C, nous avons réalisés différents programme capable de traiter un fichier CSV. Ils étaient capables de trier les données en utilisant un ABR, un AVL ou une liste chaînée au choix de l'utilisateur.

Nous avons fait appel aux fonctions fopen(), FILE en majorité. Nous avons utilisé les bibliothèques standards du C : <stdlib.h> <stdio.h> et <string.h>.

Choix du Script Shell

Notre script réalise différentes tâches afin d'assister le programme C pour trier et filtrer les données. En effet à l'aide des fonctionnalités –sort –awk ou encore –cut (pour ne citer que celles que nous avons le plus utilisées), tout cela nous a permis d'obtenir des fichiers CSV plus léger que celui d'origine.

Le Shell représente la majeure partie de notre travail dû à un problème qui nous est arrivé avec le langage C. (cf. Les difficultés rencontrées lors du projet)

Répartition du travail/planning

Pour faire avancer ce projet, nous avons mis en place un calendrier détapes. Dans un premier temps, nous avons adopté une approche théorique du projet pour bien comprendre ce que nous devions réaliser et comment le faire. Ce travail préparatoire nous a permis d'identifier les points clés et les étapes à suivre dans la mise en œuvre du programme Shell et C. Notre planning est basé sur des réunions et des semaines de travail spécifiques.

En termes d'heure de travail, nous avons travaillés pendant les heures de TD, soit 3 heures par semaine. Lorsque mes horaires académiques et personnels le permettaient, nous organisions des réunions le week-end et des soirées spécifiques pour faire avancer le projet.

Nous n'avons pas assigné de tâches spécifiques pour les honorer. Chacun travaillait et progressait à son rythme dans les différentes parties du programme. Nous n'avons pas cherché à arranger les choses tout de suite. Nous avons bien réparti le travail entre nous deux

Les difficultés rencontrées lors du projet

Shell

Pendant le processus de création du projet nous nous sommes heurtés à différentes difficultés.

La première est que le Shell est un langage nouveau pour nous. Or il y a énormément de commandes linux à connaître pour pouvoir construire quelque chose de cohérent. Nous avons donc dû faire des recherches sur Internet en amont pour prendre connaissance de ce que nous pouvions faire avec ce langage.

C

Le deuxième problème rencontré est un problème de liaison entre le C et le Shell. Nous n'avons pas réussi à faire communiquer les deux langages entre eux. Le manque de temps ne nous a pas aidés avec cette démarche et nous avons donc dû trouver une nouvelle solution. Nous avons fait le choix de trier directement le fichier d'entrée en Shell quitte à se retrouver avec des temps d'exécutions assez longs.

Après quelques optimisations nous avons réussi à faire un script en Shell relativement solide. Toutefois comme le C est demandé dans le projet nous avons mis à disposition notre travail sur cette partie dans le dossier de téléchargement

Graphiques

La dernière difficulté rencontrée est celle concernant l'affichage des graphiques. Le Shell étant un nouveau langage pour nous, utiliser GNUPLOT fût une tâche très complexe. Après maintes recherches sur le sujet nous avons fait le choix de ne pas proposer tous les types de graphique demandés. Seuls les graphiques interpolés et simples fonctionnent dans le programme.

Ressenti du groupe suite au projet

L'expérience vécue était intéressante pour notre groupe. Nous avions eu des problèmes et trouver des solutions était parfois long et fastidieux, mais au final nous avons tout fait pour rendre le projet dynamique.

Application du projet

Voici quelques exemples pour illustrer les fonctionnalités de notre programme :

```
./projet_meteo.sh -f data.csv -h -O
```

Figure 1 : Test des filtres de localisation

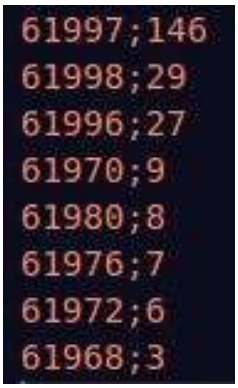


Figure 2 : Résultat du filtre –O

On retrouve bien l'ID de toutes les stations uniquement située dans l'Océan Indien (ID commençant par 61)

```
./projet_meteo.sh -f data.csv -m
```

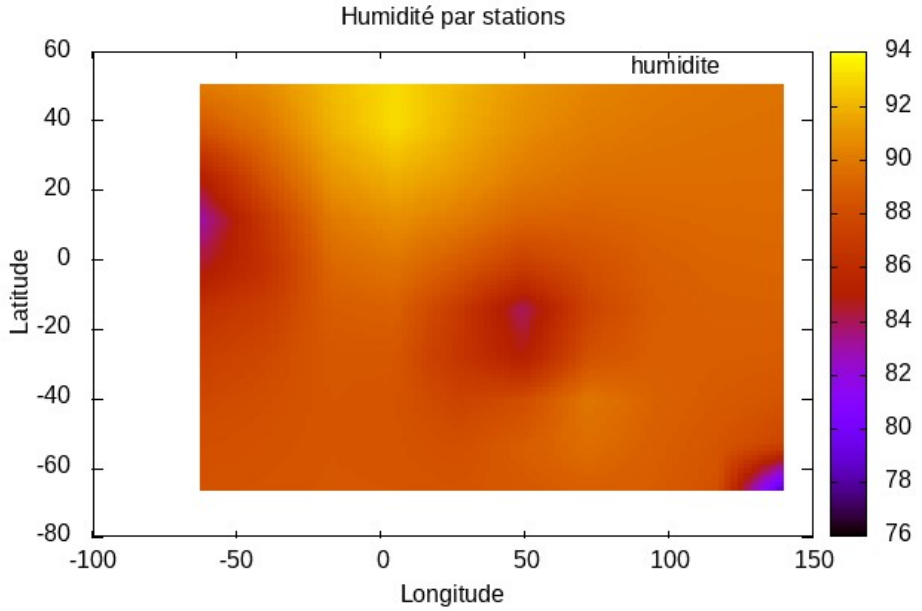


Figure 3 : Commande pour afficher l'humidité en fonction des coordonnées géographiques

```
./projet_meteo.sh -f data.csv -h
```

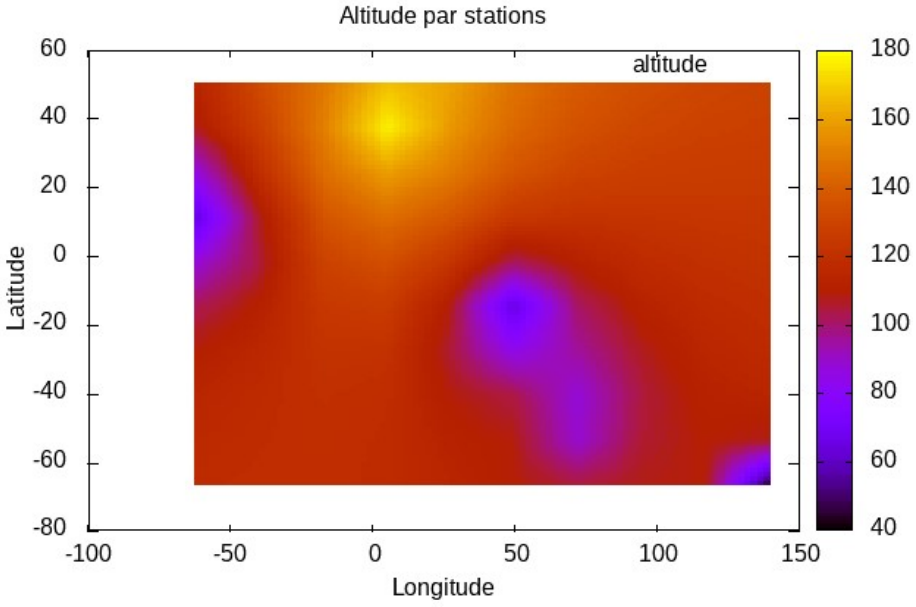


Figure 4 : Commande pour afficher l'altitude en fonction des coordonnées géographiques

```
./projet_meteo.sh -f data.csv -t2
```

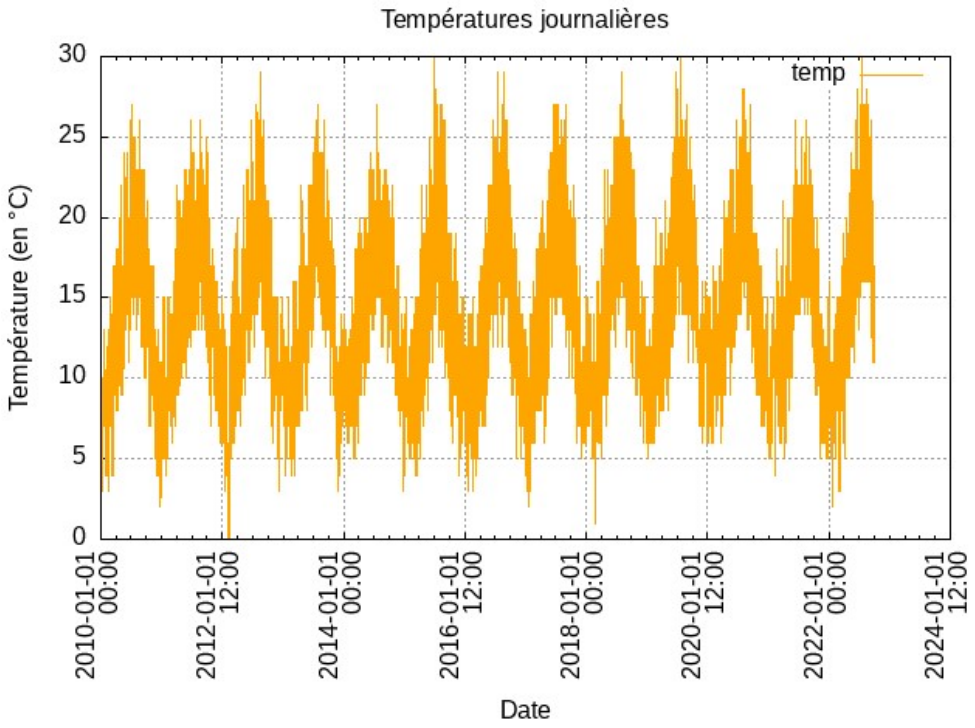


Figure 5 : Commande pour afficher les températures moyenne en fonction des dates