Contribution :

- La répartition des tâches au sein du groupe :

	Lili-Rose GICQUEL	FARRIS Théo
script shell	Les Commentaires. Option help.	Les fonctions du script shell.
langage C	Fonctions du langage C : AVL, ABR, TAB. Le fichier main. Le makeFile.	Les headers. Les commentaires.
GnuPlot	Les commentaires et les tests.	Le programme pour les graphiques du GnuPlot.

Nous avons fait ensemble le document README et le pdf.

- Le planning de réalisation :

Nous avons commencé la réalisation du programme pendant les cours durant la semaine avant les vacances de Noël.

On s'est vu plusieurs fois pendant les vacances de Noël pour avancer dans notre projet. Nous avons travaillé le script et le début du C.

Cependant, nous n'avons pas pu beaucoup avancer durant la période des partielles avec les multiples révisions que nous avons dû faire. Nous avons tout de même avancé et corrigé des erreurs dans le script shell mais aussi fait les commentaires.

Enfin, nous avons dû mettre les bouchés doubles après les partielles afin de pouvoir finir notre projet. Nous sommes restés tous les jours après les cours pour travailler le projet et à chaque moment que nous avions de disponible entre les cours ou le week-end. Nous avons fini les fonctions du script shell et le langage C. Mais aussi, nous avons rédigé le README et le document pdf présent sous vos yeux.

Pour conclure, nous avons travaillé d'arrache pieds afin de rendre notre projet et de le finaliser.

Limitations fonctionnelles :

Les limitations fonctionnelles de notre programme sont :

Nous n'avons pas pu programmer le script shell afin d'établir la limitation du fichier par la date. Nous avons codé la fonction qui permet de vérifier que la syntaxe

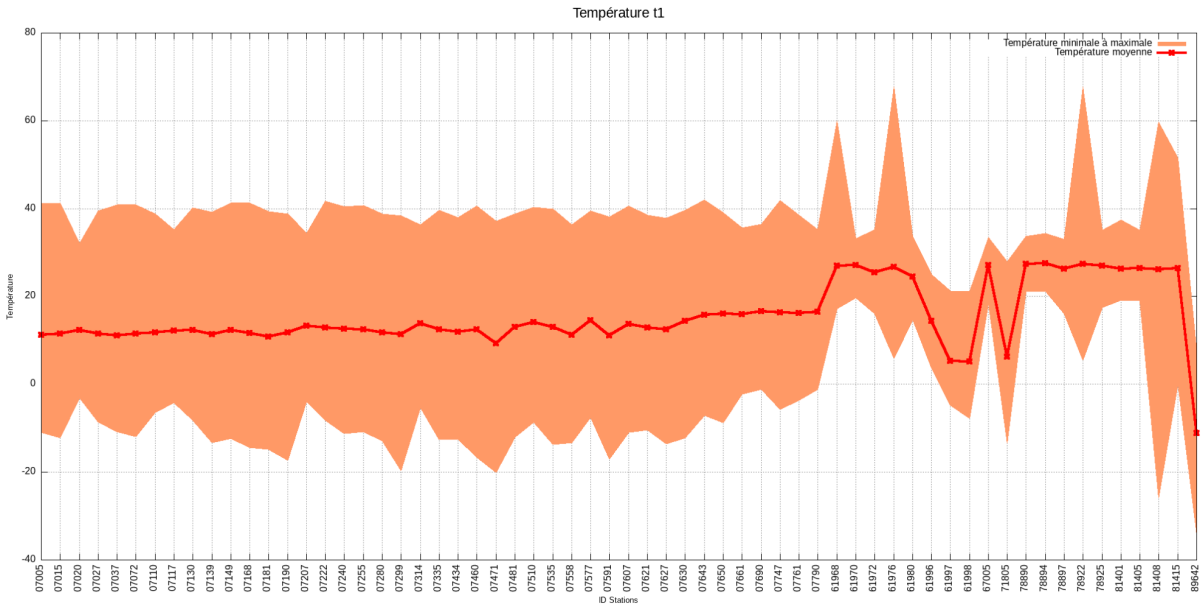
de la date est valide mais nous n'avons pas réussi à restreindre ces données à ces dates là.

De plus, nous n'avons pas pu développer le code pour le gnuplot pour les options des modes 2 et 3 fonctionnent. Nous avons créé les fonctions gnuplot, mais les données nous permettant de générer les graphiques ne sont pas bien triées.

Nous n'avons pas pu développer le langage C afin qu'il trie les données reçues par le script shell. En effet, nous n'avons pas eu le temps de permettre au langage C de trier les données qui lui sont données par le script shell.

Exemples d'exécution de notre application :

- **Exemple 1 :**

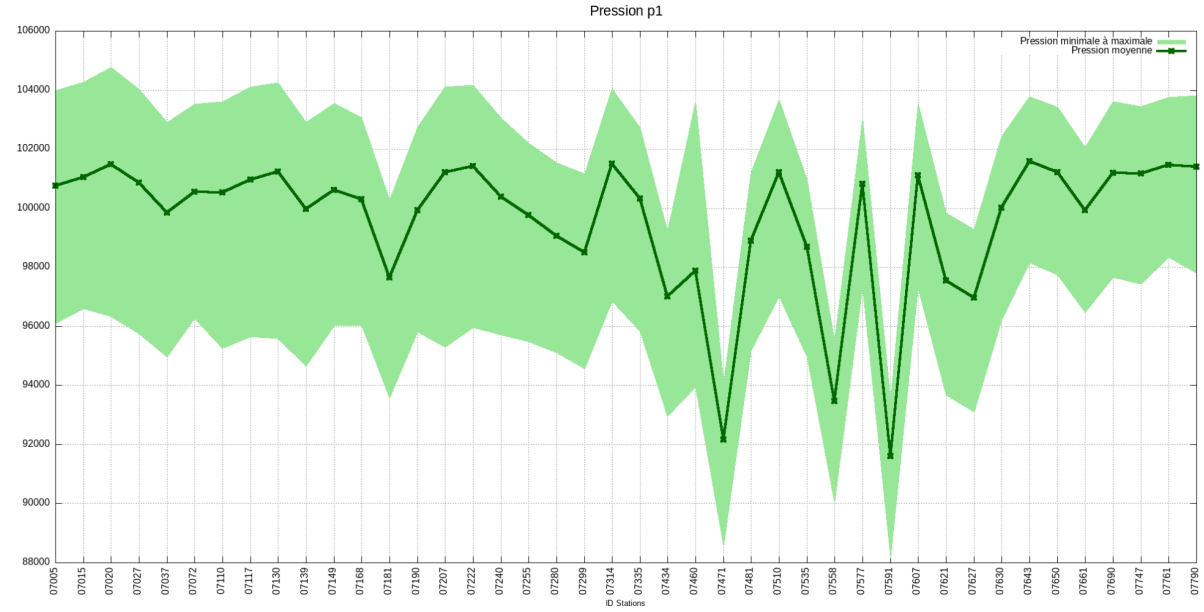


Cet exemple présente un graphique de la température en fonction de toutes les stations présentent dans le document.

La courbe rouge représente la température moyenne. Tandis que la courbe orange représente la plage de température de minimale à maximale pour chaque station.

La commande que nous avons utilisé est :
bash scriptdate.sh -f meteo_filtered_data_v1.csv -t1

- **Exemple 2 :**

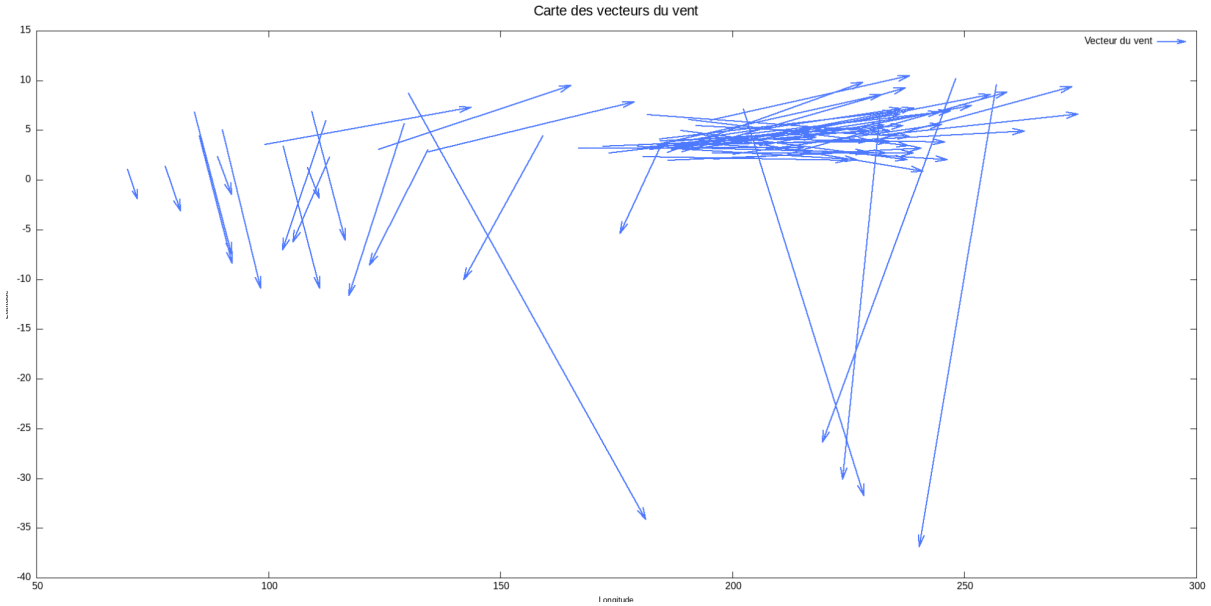


Cet exemple présente un graphique de la pression en fonction des stations en France.

La courbe verte foncée représente la pression moyenne. Tandis que la courbe verte claire représente la plage de pression minimale à maximale pour chaque station en France.

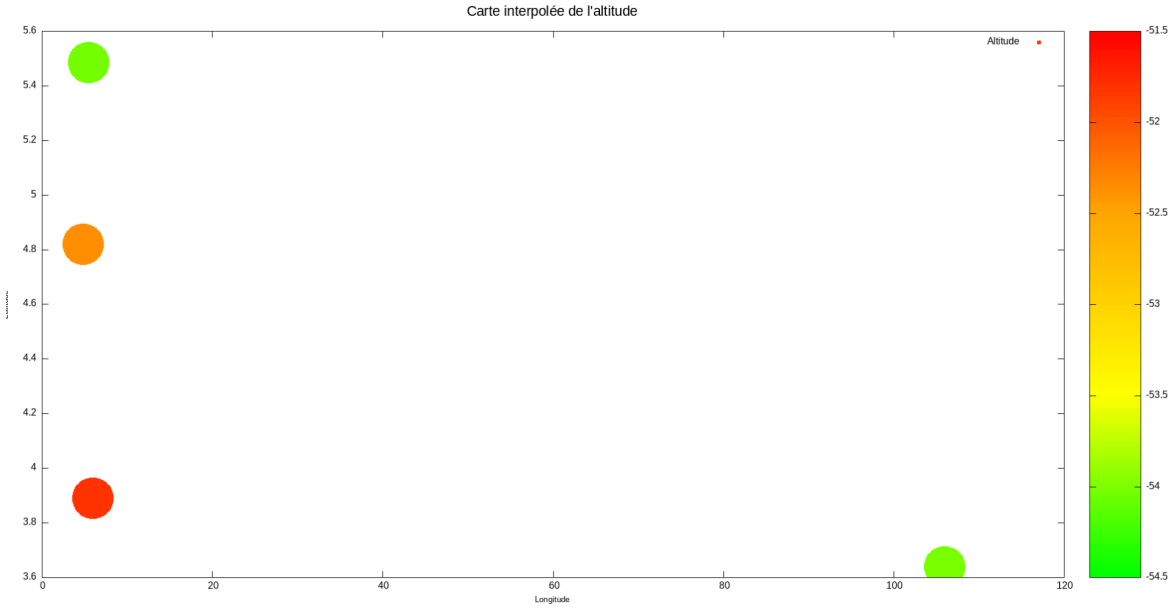
La commande que nous avons utilisé est :
bash scriptdate.sh -F -f meteo_filtered_date_v1.csv -p1

- **Exemple 3 :**



Cet exemple présente un graphique des vents en fonction de toutes les stations présentent dans le document.

Les flèches bleues représentent les vecteurs des vents pour chaque station.
La commande que nous avons utilisé est :
bash scriptdate.sh -f meteo_filtered_data_v1.csv -w



Cet exemple présente un graphique des altitudes en fonction de toutes les stations présentent dans le document.

La commande que nous avons utilisé est :
bash scriptdate.sh -G meteo_filtered_data_v1.csv -h