

Projet météo

EMESSIENE Rachel
FROMENTIN Raphaël
SEMESTRE I
Pré Ing II 2022/2023

Github : <https://github.com/lltotore/meteo>

I-Répartition des tâches

	Rachel	Raphaël
Rédaction du C	✓	✓
Rédaction du Script	✓	✓
Commentaires	✓	✓
Documentation (PDF, README)	✓	
Debugging, Cl...		✓
Formatage du code	✓	✓

III-Mise en place du projet

Pour réaliser notre projet nous avons tenu environ deux réunions de travail par semaine.

Après une première au cours de laquelle nous avons établi les différentes tâches à effectuer sous forme d'[Issues](#). Nous avons ensuite commencé à travailler chacun de notre côté. Concernant la partie C du programme, Raphaël a réalisé les fondations techniques (implémentation des différentes collections, I/O, logique de tri...) du projet tandis que Rachel a fait la logique métier (CLI, méthodes de tri, de réduction...).

II–Choix techniques

Choix globaux

Dans le programme C, nous avons utilisé un paradigme peut présent dans ce langage mais néanmoins utile pour notre projet : la programmation fonctionnelle. Le C supportant l'utilisation de pointeurs de fonctions, nous pouvions passer ces dernières en paramètre afin de réutiliser leur logique dans différentes implémentations (AVL, ABR, liste chaînée...) et ainsi réduire la quantité de code nécessaire. Si l'abstraction et la POO existaient aussi en C, la quantité de code aurait encore plus été réduite.

Choix « métier »

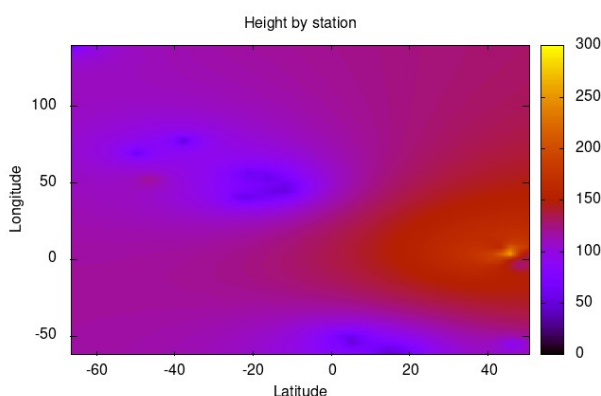
Nous avons décidé d'exclure les lignes ne contenant pas de valeur pour les colonnes utilisés (e.g Direction et vitesse du vent pour le -w). Nous avons décidé, pour le mode 2 de la température, d'utiliser les trois colonnes « ;Température (°C), Température minimale sur 24 heures (°C), Température maximale sur 24 heures (°C) ». Si une de ces trois colonnes n'est pas remplie, la ligne est ignorée.

Concernant le mode 3, les valeurs manquantes pour les heures ont été interpolées en utilisant la valeur de la mesure précédente. Les valeurs se situant après la dernière mesure d'une station pour un jour et avant la première sont nulles (0).

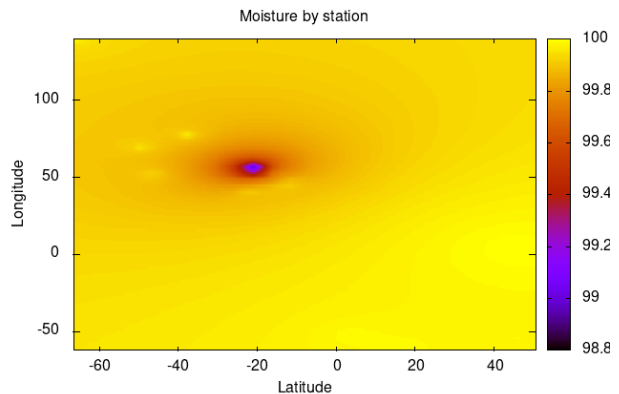
III-Limites du projet

Le projet ne supporte pas le fait de coller les valeurs à leur paramètre. Par exemple, $-t1$ n'est pas un argument valide alors que $-t 1$ l'est. Il n'est pas non plus possible de « combiner » les paramètres comme par exemple $-whm$ au lieu de $-w -h -m$.

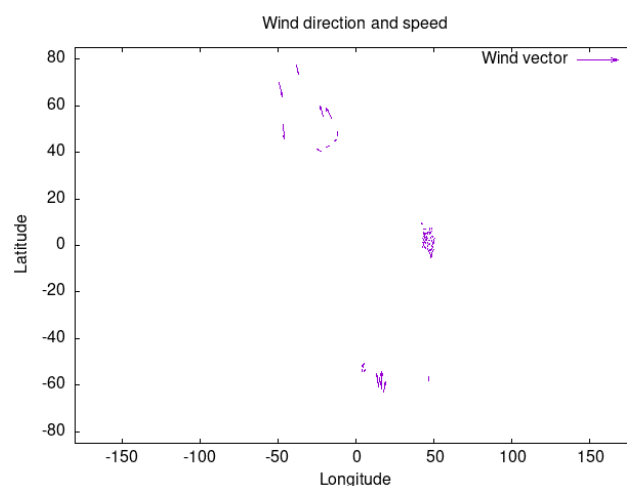
IV-Exemples de graphiques



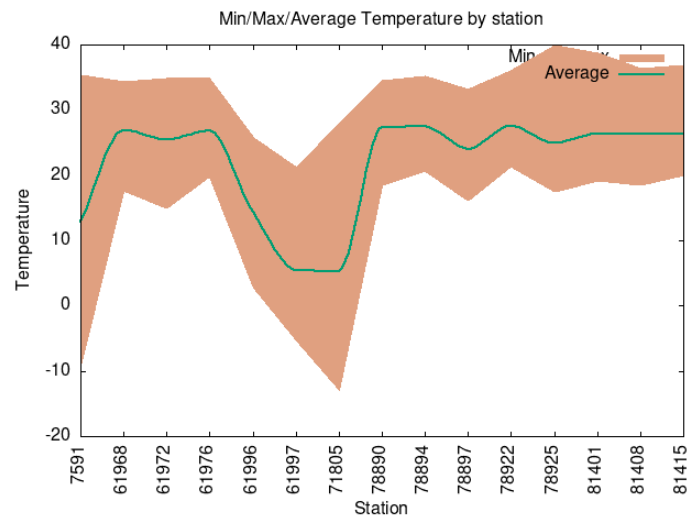
`./main.sh -f ... -h`



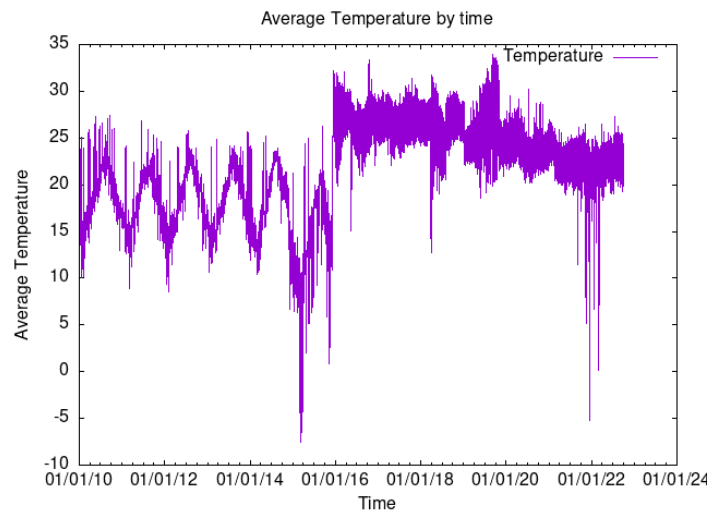
`./main.sh -f ... -w`



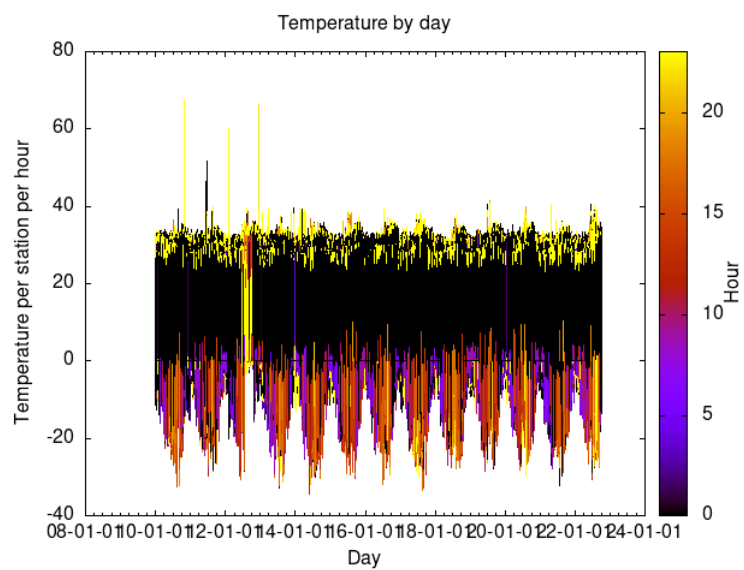
`./main.sh -f ... -w`



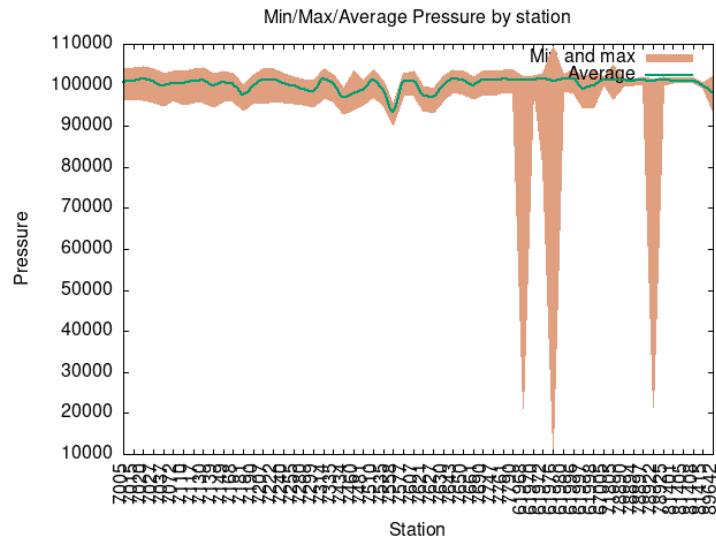
```
./main.sh -f ... -t 1
```



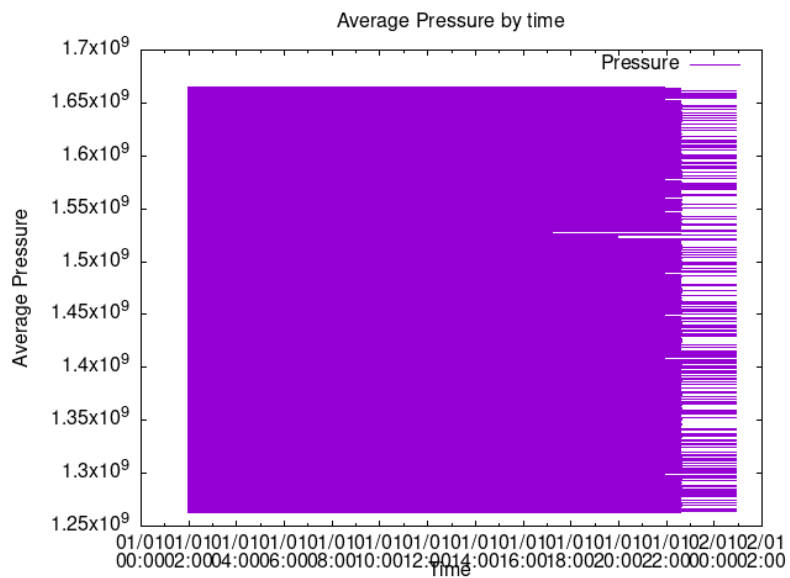
```
./main.sh -f ... -t 2
```



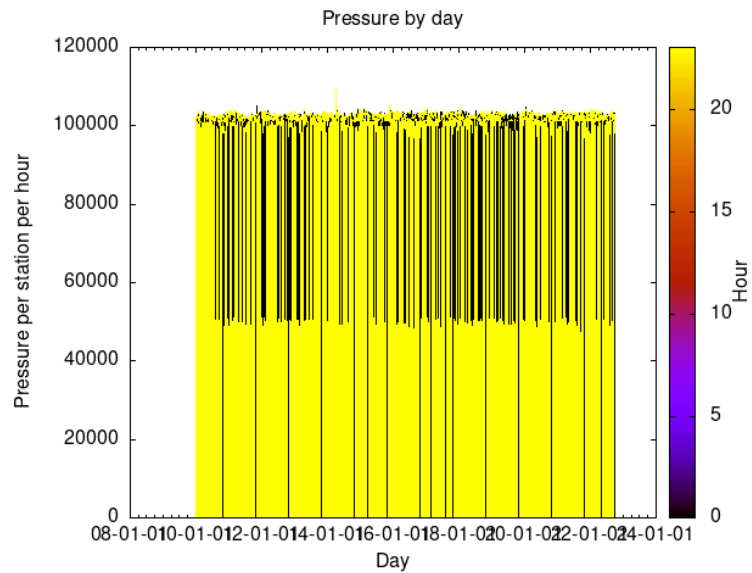
```
./main.sh -f ... -t 3
```



```
./main.sh -f ... -p 1
```



```
./main.sh -f ... -p 2
```



```
./main.sh -f ... -p 3
```