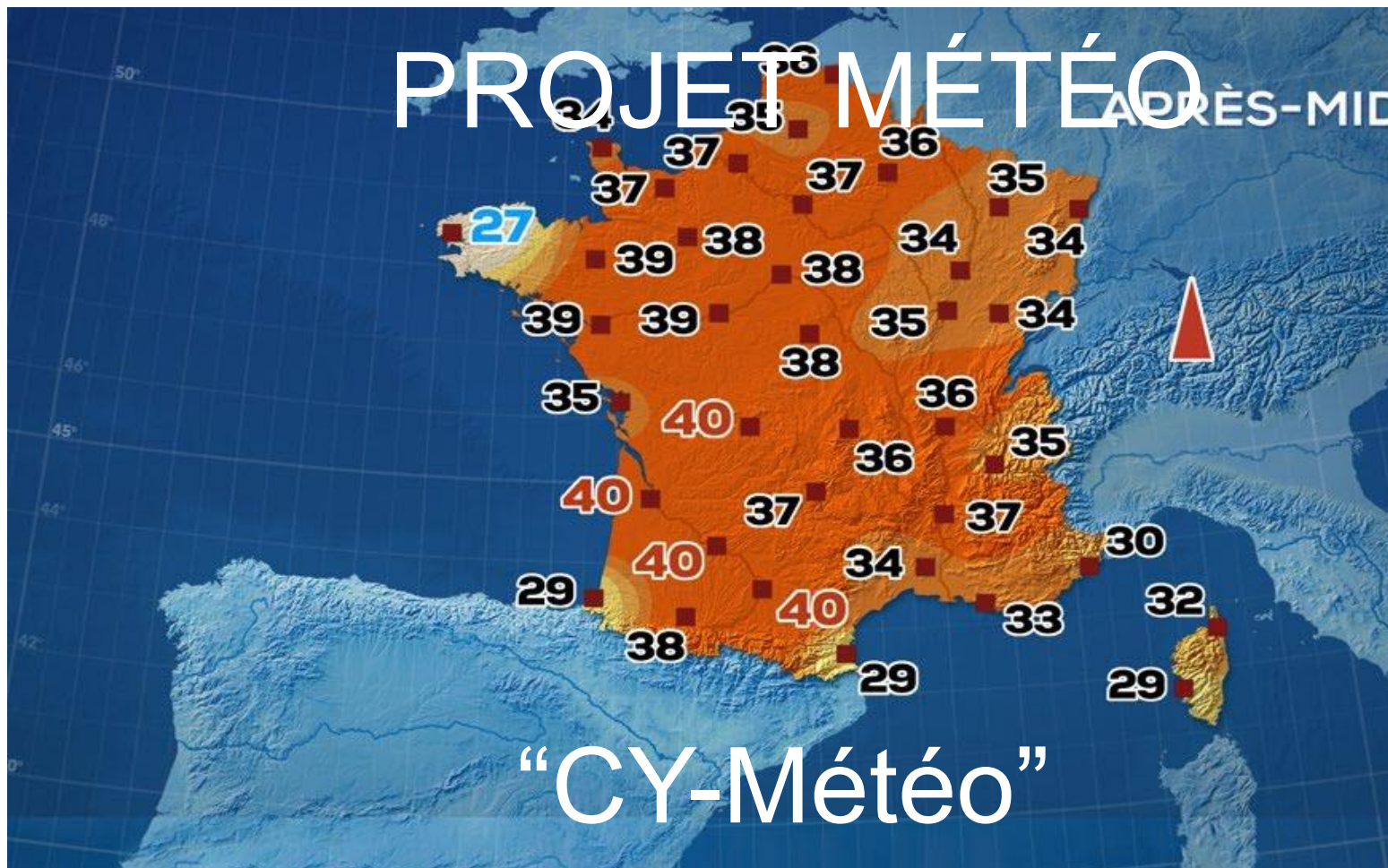


RAPPORT DE PROJET - PRÉ ING 2 MI Grp 1



Projet réalisé par

Raphaël CALHEGAS

Emma HADDAD

Projet encadré par

Romuald GRIGNON



SOMMAIRE

Introduction / Explication du sujet.....	3
Répartitions des tâches au sein du groupe, ainsi que le planning de réalisation.....	4-5
Limitations fonctionnelles (la liste de ce qui n'est pas implémentée, et/ou ce qui est implémentée mais qui ne fonctionne pas correctement).....	6
Exemples d'exécutions de notre application.....	7-13
Les images, fichiers de données d'entrée, intermédiaires et finaux.....	14
Sources.....	15

I. INTRODUCTION

Dans le cadre de notre seconde année du cycle pré-ingénieur en Mathématiques-Informatique à CY Tech, il nous est demandé un projet d'un mois et demi nous permettant de mettre en pratique nos connaissances apprises depuis notre première année. Nos diverses expériences scolaires et personnelles au cours de cette année d'études, nous ont permis d'aiguiser ce projet. Notre principale motivation était de réussir à arriver à notre objectif, mais aussi d'apprendre de nouvelles choses sur la programmation et expérimenter les différentes étapes de la conception d'un projet. Nous avons eu quelques difficultés en informatique, c'est pour cela que nous avons décidé de revoir toutes nos compétences en pratique pour réaliser un projet ambitieux. Notre motivation est forte, car nous savons que ce projet nous permettra de développer de nouvelles qualifications et d'acquérir de l'expérience.

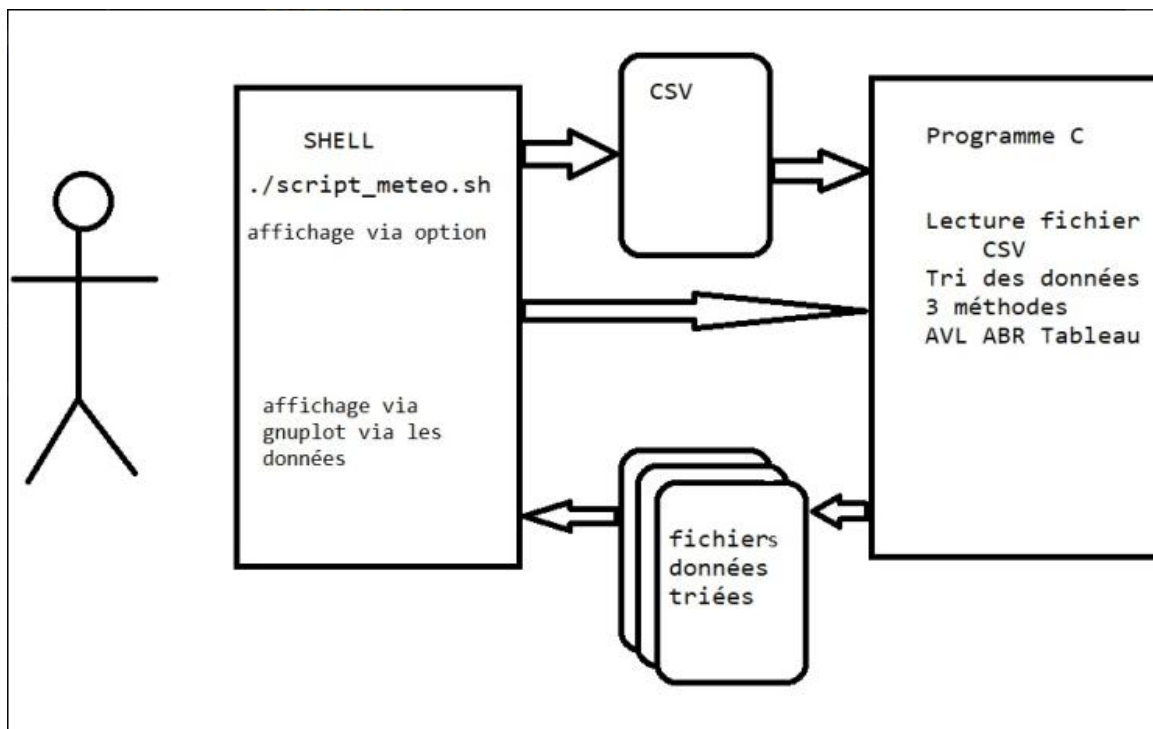
Ce projet consiste à développer le traitement des données météorologiques en Bash avec tri fait par un programme C (Tri via ABR, AVL, ou linéaire) et afficher le résultat sous forme de graphique grâce à la commande gnuplot.

L'utilisateur va lancer le programme en donnant le fichier de données en entrée et en spécifiant plusieurs arguments concernant le type de données météorologiques qu'il souhaite visualiser, la localisation, les dates, le type de structure pour trier les données (voir ci-dessous les arguments disponibles).

II. RÉPARTITION DES TÂCHES

Lors de la réception de l'énoncé, une première lecture pour survoler le sujet s'est imposée. Afin de pouvoir mieux comprendre et analyser de manière plus précise le problème demandé nous avons relu en surlignant les points importants.

Pour une compréhension plus claire nous avons créé un schéma résumant de manière globale le but final et les démarches que le programme devrait effectuer.



Suite à ce schéma, le projet s'est divisé en deux. Une partie sur le langage Shell et une autre sur le C. La partie sur le Shell a été choisie par Emma et la partie sur le C a été prise par Raphaël.

Nous nous sommes vu très régulièrement afin de pouvoir montrer notre avancée individuelle, faire des modifications, apporter de nouveaux conseils. Maintes fois, nous faisons des réunions Teams le soir pour s'aider et surtout avancer ensemble. Nous avons aussi téléchargé l'application Teamviewer qui permet de prendre le contrôle de l'ordinateur à distance en cas de difficultés.

En ce qui concerne la partie "C", nous nous sommes aidé du cours, des fonctions faites dans les exercices lors des TD et d'internet si besoin.

Afin de pouvoir trier les données nous avons utilisé trois méthodes. Le tri par tableau, le tri par arbres binaire de recherche (binary search tree) qui permet de mettre les données dans le sens que l'on souhaite (croissante, décroissante). La dernière méthode est les AVL, ce sont des ABR mais cependant équilibrés (selon la hauteur de l'arbre), permettant d'optimiser la recherche des éléments.

Nous avons tout d'abord déclaré les structures, les fonctions nécessaires (insert ABR, rotation droite, hauteur...).

Il est important de préciser que si aucune option de tri est annoncée lors de la compilation le programme triera automatiquement de manière AVL.

Pour la partie Shell, nous avons écrit un code qui compile tous les fichiers C s'ils ne sont pas déjà compilés. Le code vérifie tout d'abord si le fichier Src/main existe. Si ce n'est pas le cas, il affichera "Lancement de la compilation" et compilera les fichiers C en utilisant la commande make. Ensuite, il identifie les arguments passés à notre script. Il y a cinq types d'arguments différents : argument de données, argument géographique, argument de tri, argument de fichier et argument de date. Par la suite, il vérifie chaque argument pour voir s'il correspond à l'un des types d'arguments définis. Si c'est le cas, il l'ajoute à la liste appropriée d'arguments vrais. Et vérifie également si les dates passées en argument sont au bon format (YYYY-MM-DD HH-mm-SS) et si elles sont dans l'ordre. Si ce n'est pas le cas, il quitte le script avec un message d'erreur. Par la suite, nous avons essayé de faire en sorte que si le fichier existe et si son nom a été fourni en argument, mais aussi si le nombre d'arguments pour les différents types de raisonnement est correct. Cependant, si ce n'est pas le cas, il quittera le script avec un message d'erreur.

III. LIMITATIONS FONCTIONNELLES

- Lors du calcul de la moyenne pour la pression nous avons rencontré un problème qui peut provenir du data (dossier csv). Ce souci ne concerne qu'une vingtaine de lignes parmi toutes les lignes du document. Au lieu de prendre le minimum, le programme prend 0 comme valeur (car pour certaines lignes au lieu que ca soit un "." c'est une ",") ce qui fausse les résultats attendus.

Tandis que pour la température nous n'avons pas eu ce problème.

- Notre projet n'est pas complet car nous n'avons pas réussi à faire les fonctions "-t3" et "-p3" qui consistaient à produire en sortie les températures/pressions par date/heure par station.

- Durant l'avancement du projet nous sommes restés bloqués car pour les dates le format écrit dans le data (fichier csv) ne correspondait pas au format demandé dans l'énoncé. Nous devions suivre le modèle suivant : YYYY-MM-DD or dans le fichier csv le format était le suivant :

```
ID OMM station;Date;Pression au niveau mer;  
07558;2010-01-05T10:00:00+01:00;100280;260;
```

Après plusieurs recherches nous avons utilisé la fonction gensub qui permet de mettre les dates dans un format accepté par mktime. Mkttime nous prend une date en entrée et nous renvoie le nombre de secondes écoulé depuis le 1er janvier 1970 à 00h00.

L'utilisation de ces deux fonctions nous permet de comparer les dates et de les utiliser comme clé (key) pour les fonctions de tri.

```
;a=mktime(gensub(/(....)-(..)-(..)T(..):(..):(..)([+-.]*)/, f, "g", $2));if(a>inf && a<sup) {print $0}}'  
;a=mktime(gensub(/(....)-(..)-(..)T(..):(..):(..)([+-.]*)/, f, "g", $2));if(a>inf && a<sup) {print $0}}'
```

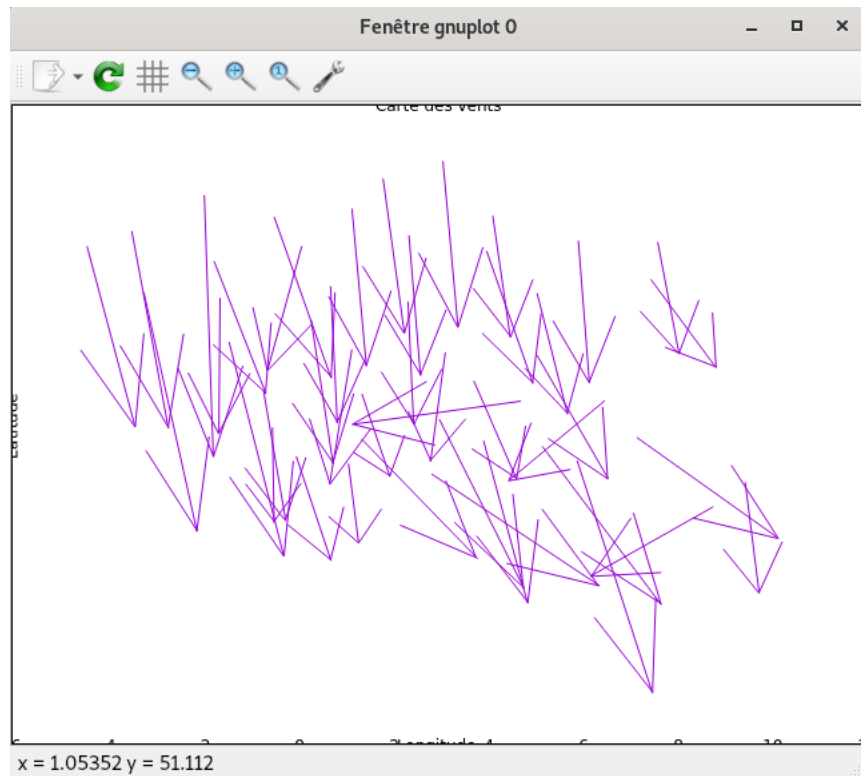
IV. EXEMPLES D'APPLICATIONS

```
cy@VBDebian1164: ~/Bureau/Projet_info-main
cy@VBDebian1164:~/Bureau/Projet_info-main$ ./meteo.sh -f meteo.csv -w -F
Filtrage géographique en cours...
Affichage de la carte des vents

Warning: slow font initializationcy@VBDebian1164:~/Bureau/Projet_info-main$
```

Commande pour compiler la carte des vents en France

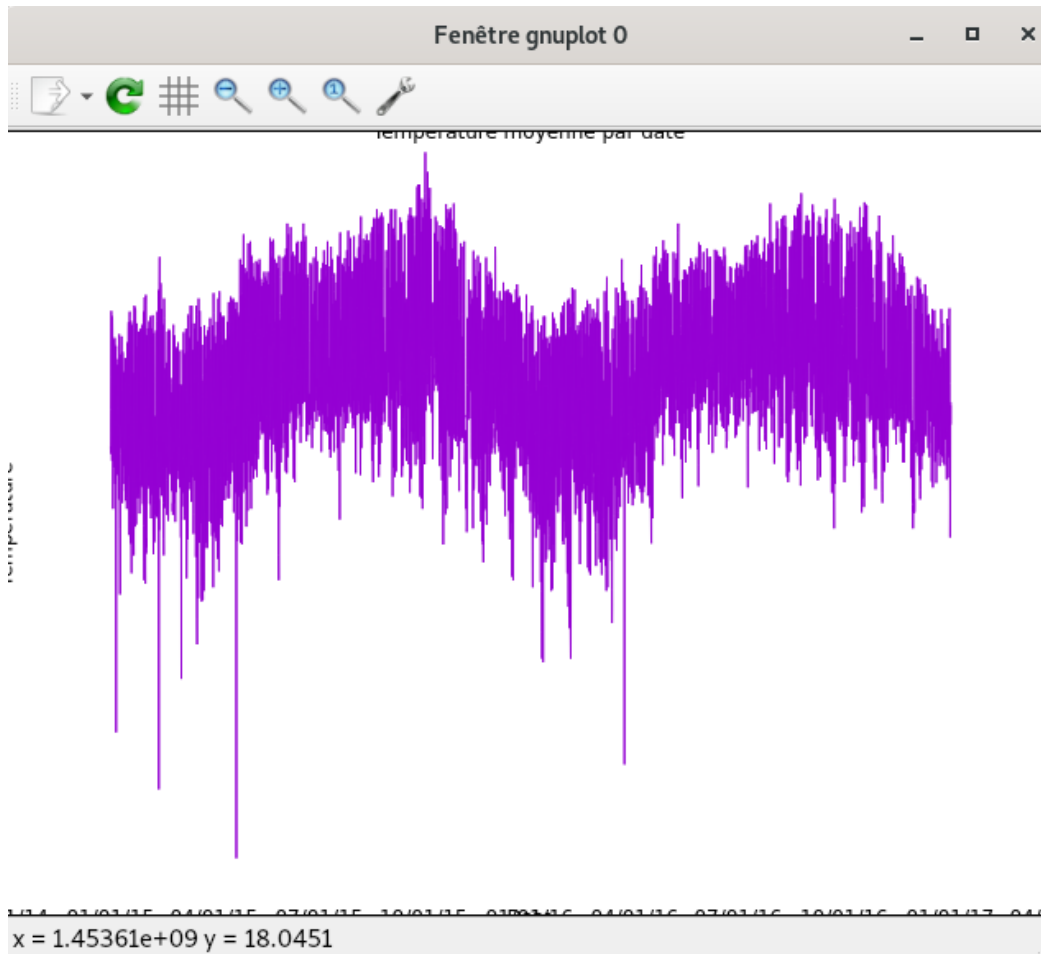
Carte des vents de la France



```
cy@VBDebian1164:~/Bureau/Projet_info-main$ ./meteo.sh -f meteo.csv -t2 --abr -A
-d 2015-01-01 2017-01-01
Filtrage géographique en cours...
Filtrage temporel en cours...
Affichage de la carte temperature moyenne par date

Warning: slow font initializationcy@VBDebian1164:~/Bureau/Projet_info-main$
```

Commande pour compiler la carte des températures aux Antilles



Production des températures moyennes par date (-t2) aux antilles (-A) entre le 1er Janvier 2015 & le 1er Janvier 2017 en utilisant un tri ABR

```
cy@VBDebian1164:~/Bureau/Projet_info-main$ ./meteo.sh -f meteo.csv -m -A --abr -
d 2015-01-01 2014-01-01

Attention ! 2015-01-01 est après 2014-01-01. Les dates doivent être cohérente
cy@VBDebian1164:~/Bureau/Projet_info-main$
```

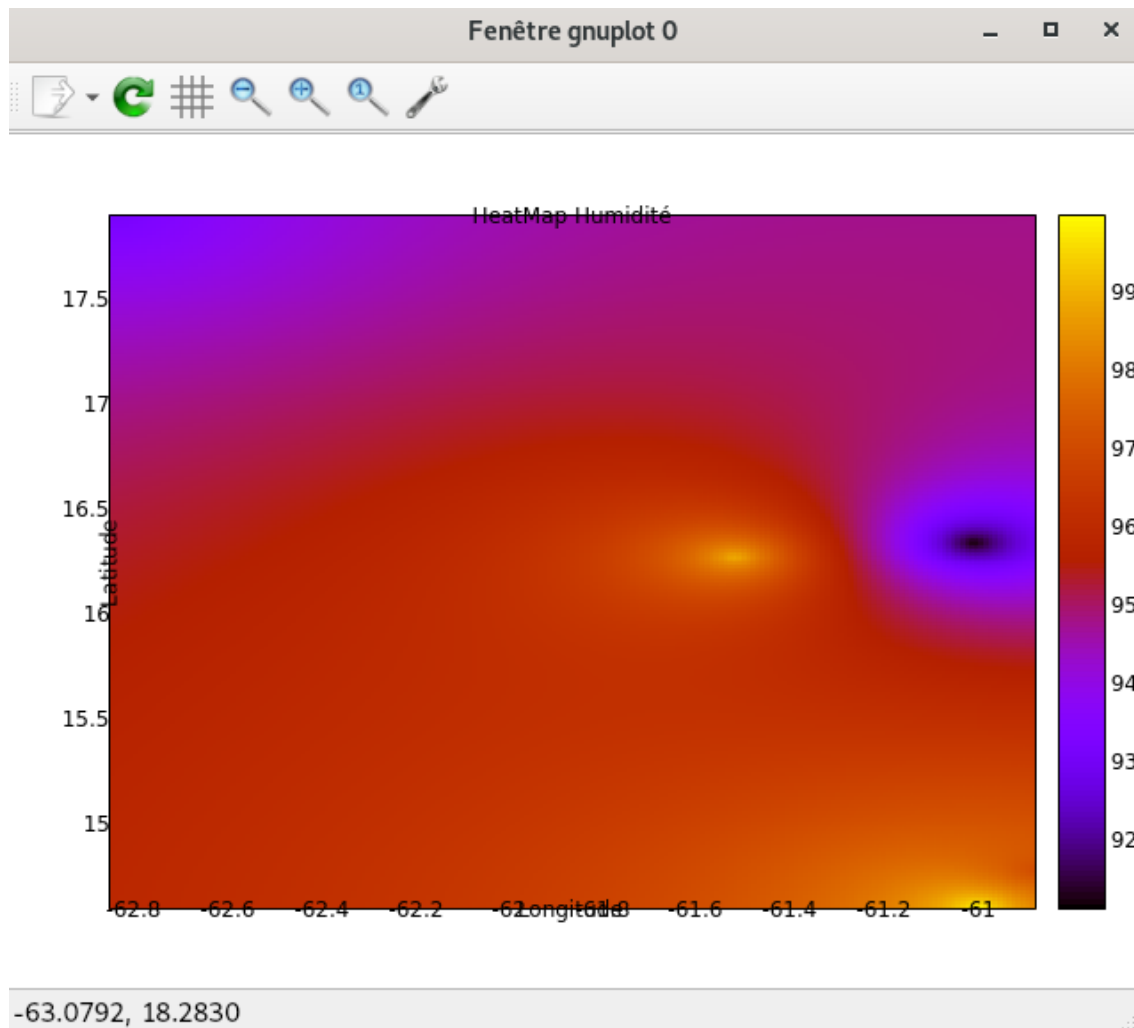
Exemple d'erreur de commande, les dates inscrites ne sont pas dans le bon ordre (-d<min> <max>)

```
cy@VBDebian1164:~/Bureau/Projet_info-main$ ./meteo.sh -f meteo.csv -m -A --abr -
d 2015-01-01 2017-01-01

Filtrage géographique en cours...
Filtrage temporel en cours...
Affichage de la carte humidite

Warning: slow font initializationcy@VBDebian1164:~/Bureau/Projet_info-main$
```

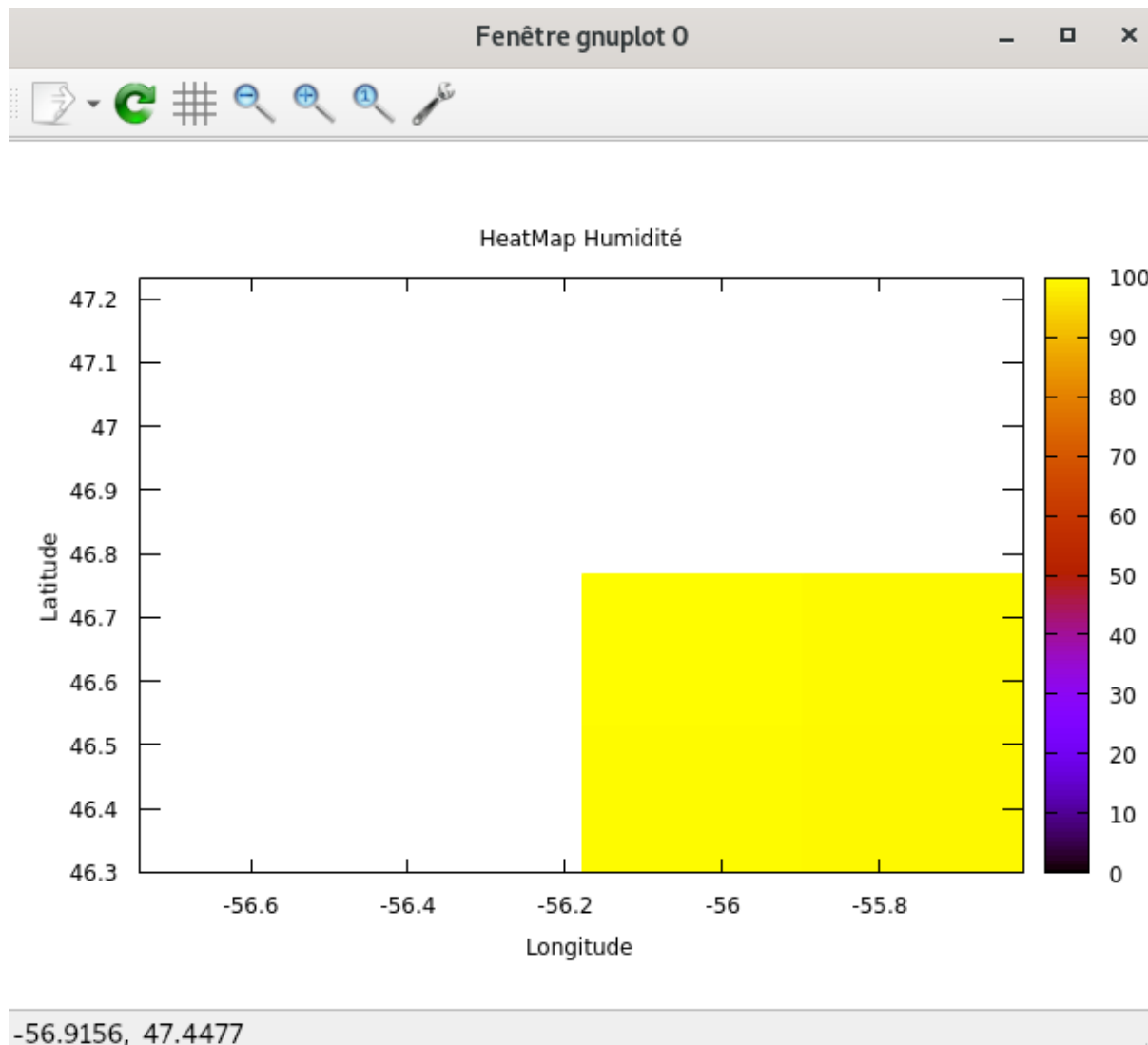
Commande pour compiler la carte d'humidité aux Antilles



Produit en sortie l'humidité entre 2015-01-01 et 2017-01-01 pour les Antilles avec un tri ABR

```
cy@VBDebian1164: ~/Bureau/Projet_info-main
cy@VBDebian1164:~/Bureau/Projet_info-main$ ./meteo.sh -f meteo.csv -S -m
Filtrage géographique en cours...
Affichage de la carte humidite
Warning: empty x range [-56.1792:-56.1792], adjusting to [-56.741:-55.6174]
Warning: empty y range [46.7663:46.7663], adjusting to [46.2987:47.234]
cy@VBDebian1164:~/Bureau/Projet_info-main$
```

Commande pour compiler la carte d'humidité à Saint-Pierre et Miquelon



Produit en sortie l'humidité maximale pour Saint-Pierre et Miquelon
Nous remarquons que toute la map n'est pas remplie mais seulement une partie. Gnuplot nous borne les données entre la longitude et la latitude. Or pour Saint-Pierre et Miquelon nous n'avons qu'une seule station déclarée. Nous n'arrivons donc pas à borner car pour borner il faut au moins 2 et ici nous n'en avons que 1.
Le carré blanc signifie les endroits où nous n'avons pas de données.

La commande help

```
cy@VBDebian1164: ~/Bureau/Projet_info-main
cy@VBDebian1164:~/Bureau/Projet_info-main$ ./meteo.sh --help
cy@VBDebian1164:~/Bureau/Projet_info-main$
```

Commande pour compiler

```
cy@VBDebian1164: ~/Bureau/Projet_info-main
# Introduction
Developpement d'un projet de traitement de données météorologiques en Bash avec
tri fait par un programme C (Tri via ABR, AVL, ou lineaire).

# Objectif du Projet
L'utilisateur va lancer le programme en donnant le fichier de données en entrée
et en spécifiant plusieurs arguments concernant le type de données météorologi
es qu'il souhaite visualiser, la localisation, les dates, le type de structure p
our trier les données (voir ci-dessous les arguments disponibles)

# Les arguments :
## Type de données météorologiques
Ces options ne sont pas exclusives entre elles. Si on choisit plusieurs types de
données, on aura plusieurs graphiques.
- Les options -t et -p doivent être accompagnées d'un mode :
  -t<mode> : (t)emperatures.
  -p<mode> : (p)ressions atmosphériques.
- Pour ces 2 options, il faut indiquer la valeur du <mode> :
  - 1 : produit en sortie les températures (ou pressions) minimales, maximales e
t moyennes par station dans l'ordre croissant du numéro de station.
  - 2 : produit en sortie les températures (ou pressions) moyennes par date/heur
e, triées dans l'ordre chronologique. La moyenne se fait sur toutes les stations
```

```
Il est aussi nécessaire d'avoir la fonction gawk à jour. Si ce n'est pas le cas,
téléchargez-la à l'aide de la commande suivante
...
sudo apt-get install gawk
...
Production des altitudes (-h) & des cartes de vents (-w) en France (-F) sur tout
e la période
...
./meteo.sh -w -F -f data.csv
...
Production des temperatures moyennes par date (-t2) aux antilles (-A) entre le
1er Janvier 2015 & le 1er Janvier 2017 en utilisant un tri ABR
...
./meteo.sh -A -f data.csv -t2 --abr -d 2015-01-01 2017-01-01
...
Cependant, si vous rencontrez des difficultés pour compiler aidez-vous de l'opti
on suivante
...
./meteo.sh --help
...
Elle vous permettra d'obtenir l'affichage d'un aide détaillé à l'utilisation du
script. Dès que vous n'en aurez plus l'utilité, il vous suffira d'appuyer sur la
touche "q" de votre clavier pour recompiler
(END)
```

```
Il est aussi nécessaire d'avoir la fonction gawk à jour. Si ce n'est pas le cas,
téléchargez-la à l'aide de la commande suivante
```
sudo apt-get install gawk
```
Production des altitudes (-h) & des cartes de vents (-w) en France (-F) sur toute la période
```
./meteo.sh -w -F -f data.csv
```
Production des températures moyennes par date (-t2) aux antilles (-A) entre le 1er Janvier 2015 & le 1er Janvier 2017 en utilisant un tri ABR
```
./meteo.sh -A -f data.csv -t2 --abr -d 2015-01-01 2017-01-01
```
Cependant, si vous rencontrez des difficultés pour compiler aidez-vous de l'option suivante
```
./meteo.sh --help
```
Elle vous permettra d'obtenir l'affichage d'un aide détaillé à l'utilisation du script. Dès que vous n'en aurez plus l'utilité, il vous suffira d'appuyer sur la touche "q" de votre clavier pour recompiler
(END)
```

Quelques captures du Readme avec les commandes importantes et la commande de fin pour pouvoir recompiler à la suite.

Voici quelques commandes permettant de compiler:

Pensez à mettre le dossier data.csv dans le même répertoire que les fichiers du projet lors de la compilation

`./meteo.sh -f data.csv -w` : carte des vents pour toutes les stations du projet

`./meteo.sh -f data.csv -w -A`: carte des vents pour les Antilles

`./meteo.sh -f data.csv -w -F`: carte des vents pour la France
ceci est valable pour toutes les stations (il faut juste mettre -lettre de la station voulu)

`./meteo.sh -f data.csv -h` : carte de l'altitude pour toutes les stations du projet

`./meteo.sh -f data.csv -h -Q`: carte des altitudes pour l'Antarctique
ceci est valable pour toutes les stations (il faut juste mettre -lettre de la station voulu)

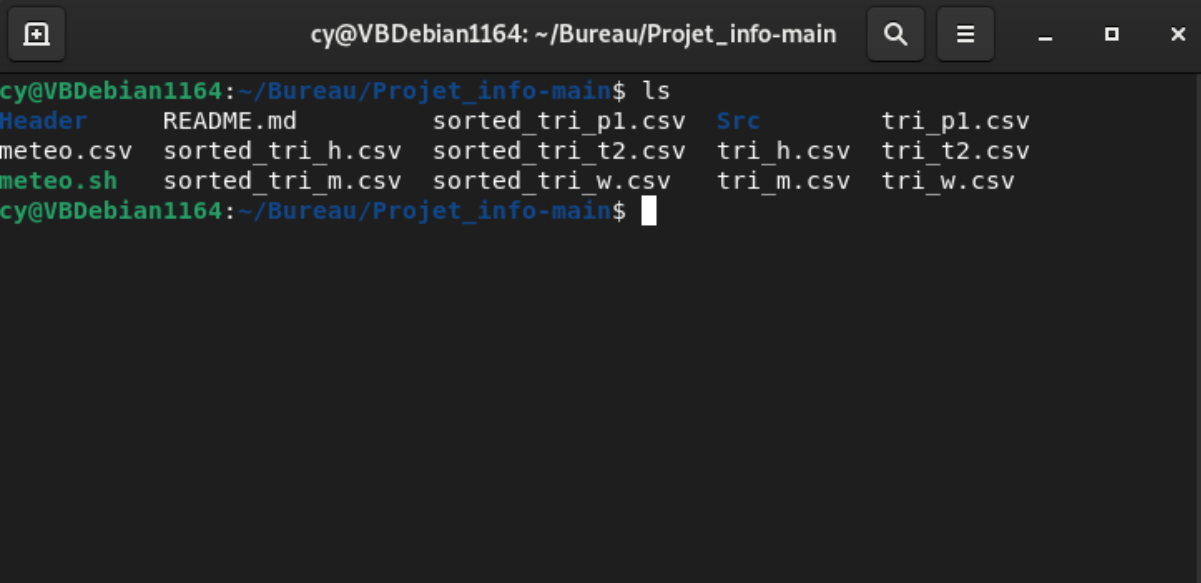
`./meteo.sh -f data.csv -m` : carte de l'humidité pour toutes les stations du projet

`./meteo.sh -f data.csv -p1 -d<min><max>` : Pression minimales, maximales et moyennes pour toutes les stations entre telle et telle dates choisies

`./meteo.sh -f data.csv -t2 -d 2015-01-01 2018-01-01 -G` : Température moyennes par dates/heures pour la Guyane Française entre le 1er janvier 2015 et le 1er janvier 2018

`./meteo.sh -f data.csv -p2 -d 2016-03-27 2018-04-06 --avl -O` : Pressions moyennes par dates/heures pour l'Océan indien entre le 27 mars 2016 et le 6 avril 2018 en utilisant un tri AVL

V. IMAGES, FICHER D'ENTRÉES/FINAUX



```
cy@VBDebian1164: ~/Bureau/Projet_info-main
cy@VBDebian1164:~/Bureau/Projet_info-main$ ls
Header      README.md      sorted_tri_p1.csv  Src          tri_p1.csv
meteo.csv   sorted_tri_h.csv sorted_tri_t2.csv  tri_h.csv    tri_t2.csv
meteo.sh    sorted_tri_m.csv sorted_tri_w.csv   tri_m.csv    tri_w.csv
cy@VBDebian1164:~/Bureau/Projet_info-main$
```

Grâce au terminal nous pouvons voir deux fichiers : ceux d'entrée et ceux de sortie.

Les fichiers d'entrées sont les tri_"...."csv

Les fichiers de sorties sont les sorted_tri_"...."csv

VI.SOURCES

-https://www.google.com/search?q=fond+m%C3%A9t%C3%A9+france&tbm=isch&ved=2ahUKEwih n6W6n-38AhVip0wKHSZMBmUQ2-cCegQIABAA&oq=fond+m%C3%A9t%C3%A9+france&gs_lcp=CgNpbWcQAzoECCMQJzoGCAAQCBAeOgQIABAEUMQHWJIGYIhlaAZwAHgAgAFSiAHaBZIBAJE0mAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=AaXWY-GaG-LOsgKmmJmoBg&bih=927&biw=1920&client=firefox-b-d#imgsrc=BI7NasL_8KP6uM

-<https://koor.fr/C/cstdio/fgets.wp>

-<https://koor.fr/C/cstdlib/atoi.wp>

-Cours et TD de l'année

-https://fr.wikipedia.org/wiki/CY_Tech#/media/Fichier:CY_Tech.svg

-<https://stackoverflow.com/>