

CY-TRAUCH

Logiciel de traitement de données excel avec interface

Sommaire

1) Explication du programme	3
- Traitement D1	
- Traitement D2	
- Traitement L	
- Traitement T	
- Traitement S	
- Traitement Bonus	
2) Organisation du groupe et Planning	7
3) Limitation et difficultés	8
4) Exécution et Programme	9

1) Explication du programme:

Notre programme CY-Truck est un programme en shell et en C. A l'intermédiaire de données, l'objectif de notre programme est de faire des graphiques de ces statistiques en fonction des demandes de l'utilisateur (avec tous les différents traitements). Ces différents traitements sont listés ci-dessous.



Il ya **2** versions

1 avec interface

1 sans interface

Ce programme fonctionne donc de deux façons différentes:

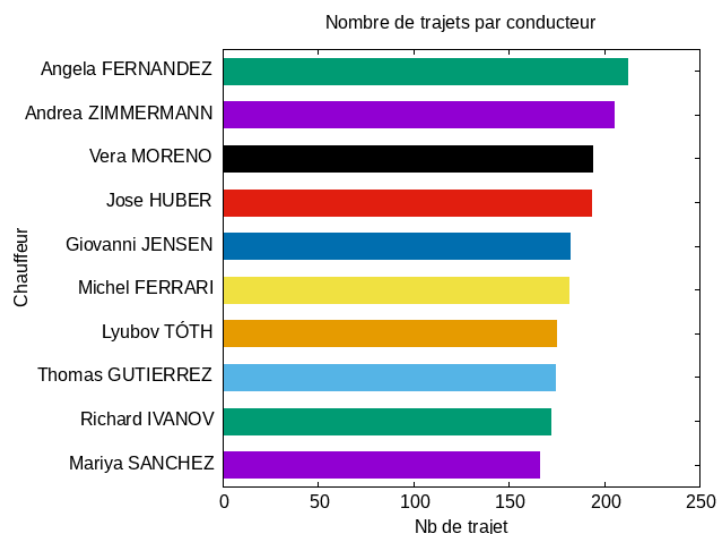
- **La première façon est de le lancer avec des arguments**, en indiquant le chemin complet d'où se trouve data.csv puis les traitements qu'on souhaite lancé.
 - **La deuxième façon est de le lancer sans arguments**, et une **interface utilisateur** va s'afficher pour lancer les traitements souhaités.
- Pour plus d'informations sur l'utilisation, consulter le readme.

Bien s'assurer également que le fichier data.csv est dans le dossier data.

Ce logiciel propose différents traitements:

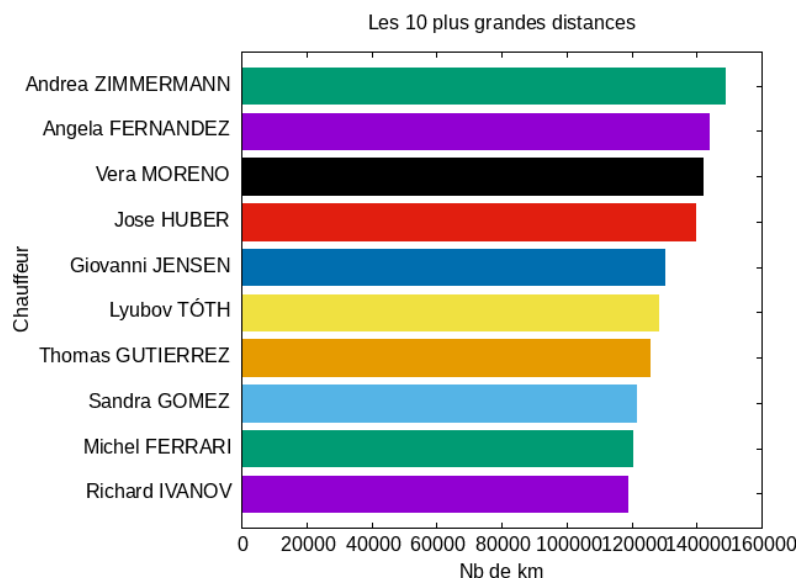
Traitement D1: Conducteurs avec le plus de trajets : option -d1

Ce traitement crée un graphique de type histogramme horizontal avec en ordonnée les noms complets des conducteurs et en abscisse le nombre de trajets effectués. Ce traitement a été réalisé à l'aide d'un script Shell.



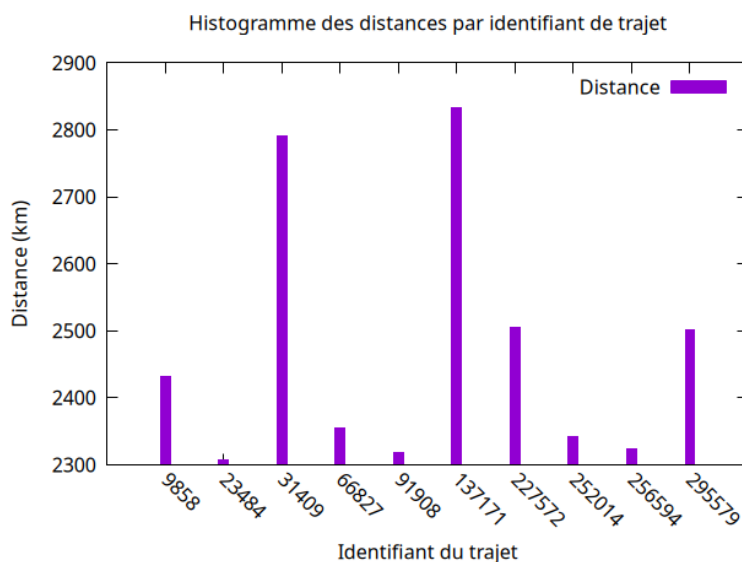
Traitement D2: Conducteurs et la plus grande distance: option -d2

Ce traitement affiche un histogramme de type horizontal qui affiche les dix plus grandes distances parcourues par conducteurs, avec en ordonnée les noms des conducteurs et en abscisse la distance totale parcourue. Ce traitement a été réalisé à l'aide d'un script Shell.



Traitement L: Les 10 trajets les plus longs : option -l

Ce traitement récupère la distance totale de chaque trajet (la somme de toutes les étapes) et ordonne ce résultat par numéro d'identifiant de trajet croissant. A partir de cela, il va créer un graphique de type histogramme vertical qui affiche les 10 distances les plus grandes avec en abscisse l'identifiant du trajet et en ordonné la distance en km. Ce traitement a été réalisé à l'aide d'un script Shell.



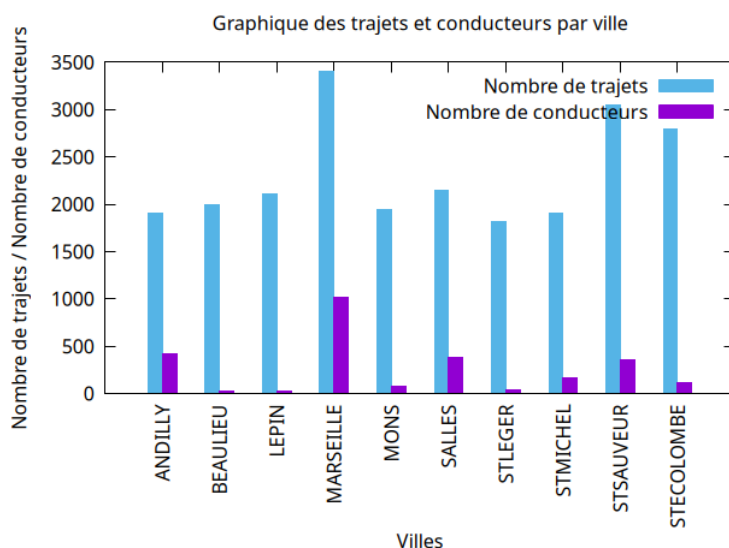
Traitement T: Les 10 villes les plus traversées : option -t

Ce traitement compte le nombre de trajets qui parcourent chaque ville ainsi que le nombre de fois où ces villes ont été des villes de départ de trajets et ne conserve que les dix premières villes avec le plus de trajets par ordre alphabétique.

Le programme va donc créer un graphique de type histogramme avec en abscisse le nom des villes et en ordonnée le nombre de trajets.

Pour chaque ville, il y a 2 barres verticales, celle de gauche indique le nombre de trajets total qui traversent cette ville, et celle de droite le nombre de fois où cette ville est la ville de départ d'un trajet.

Ce programme utilise du script Shell ainsi que du C.



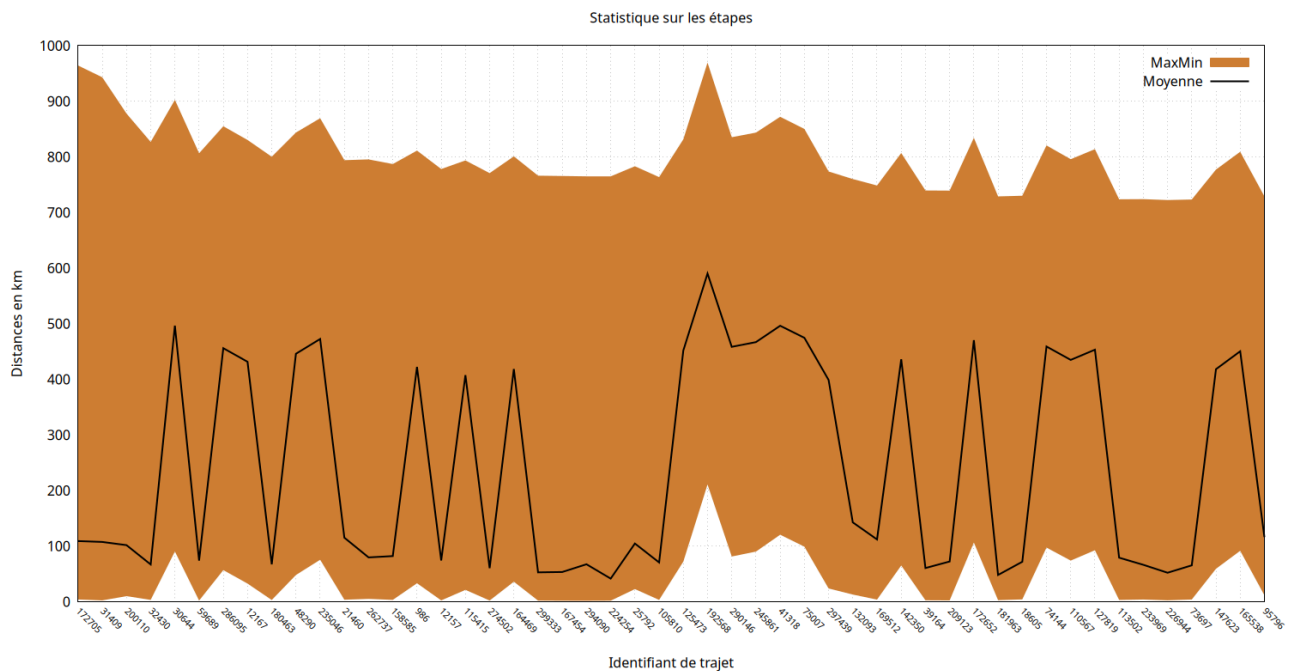
Traitement S: Statistiques sur les étapes : option -s

Ce traitement récupère les distances minimales, maximales et moyennes des étapes pour chaque trajet.

Il va créer un graphique de type courbes min-max-moyenne, avec deux courbes indiquant les valeurs min et max des distances et une troisième entre les deux pour indiquer la moyenne.

Les données sont triées par ordre décroissant et gardent les 50 premières valeurs.

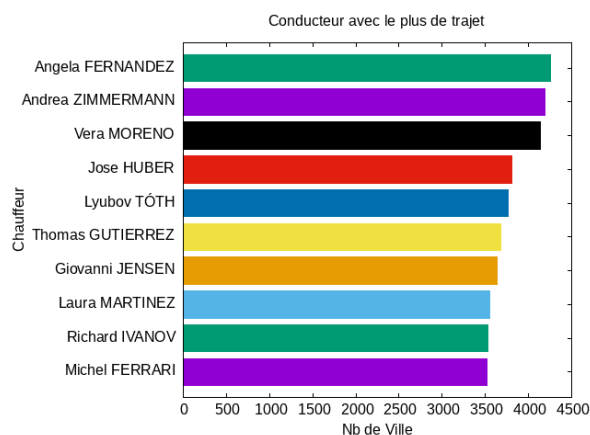
Il a été réalisé à l'aide d'un script Shell ainsi que du C.



Traitement Bonus: Conducteur avec le plus de trajet

Ce traitement récupère le nombre de fois qu'un chauffeur traverse une ville. C'est un traitement bonus complet et bien intégré à notre projet CY-Truck.

Ce dernier veille bien à supprimer les doublons et qui va créer un graphique de type histogramme horizontal avec en ordonnée les noms des conducteurs et en abscisse le nombre de ville parcourue. Ce traitement a été réalisé à l'aide d'un script Shell.



2) Organisation du groupe et Planning

Dès que le sujet du projet a été lancé, nous avons décidé de nous répartir les différents traitements.

Pour l'interface graphique, Ilyass a en partie reprise celle qu'il avait réalisée pour son projet de pré-ing1.

Comme il y avait trois différents traitements en shell et deux traitements principalement en c, nous les avons répartis de la manière suivante : un traitement en shell par personne (traitement d1 réalisé par Gabriel, traitement d2 par Eléonore et traitement L par Ilyass). En ce qui concerne les traitements avec du c, Eléonore et Gabriel ont réalisé le traitement t et Ilyass le traitement s.

Nous avons comme objectif de réaliser les cinq traitements en moins de deux semaines, ce qu'on a plus ou moins réussi à faire. Il nous a donc fallu une très bonne organisation. Ce n'est pas moins de beaucoup d'entre aides et de réunions (environ trois par semaines) qui nous ont permis de terminer nos traitements selon notre objectif.

Cependant, le sujet a été modifié plusieurs fois, c'est-à-dire que même si on avait fait le traitement initialement demandé, nous avons dû en refaire certains plusieurs fois.

La semaine des partiels arrivant, nous avons mis de côté le projet et l'avons repris dès la fin des partiels, en refaisant les traitements C et les améliorant en prenant en compte le nouveau cahier des charges demandés.

C'est donc lors de la semaine de remise que nous avons fait le travail qu'il resait à faire : refaire le traitement T avec les nouvelles valeurs de M.Romuald (Ilyass), écrire certains commentaires (faits par Gabriel et Eléonore), partitionner nos codes (par Ilyass), écrire le MakeFile (par Ilyass), le pdf explicatif (Eléonore), et la fonction help (Gabriel). Tout cela a été terminé le jeudi 1^{er} février pour nous laisser une journée de vérification de potentiels problèmes.

Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 5	Semaine 6	Semaine 7	Semaine 8
Ilyass: -traitement L -traitement S Gabriel: -traitement d1 Eleonore: -traitement d2	Ilyass: -Interface -traitement S Gabriel: -traitement d1 Eleonore: -traitement d2	Ilyass: -Interface -traitement S Gabriel: traitement T Eleonore: traitement T	Ilyass: -traitement S Gabriel: traitement T Eleonore: traitement T	Ilyass: -Interface makefile Gabriel: commentaire Eleonore: commentaire	Ilyass: traitement T -bonus Gabriel: indentation Eleonore: indentation	Partiel	Ilyass: -Interface -sécurité Gabriel: -vérification Eleonore: -vérification

--	--	--	--	--	--	--	--

Nos sources:

Il est également important de mentionner que ne connaissant par gnuplot, nous nous sommes appuyés sur le site « Stackoverflow » pour nous aider à réaliser nos graphiques

(<https://stackoverflow.com/questions/62848395/horizontal-bar-chart-in-gnuplot>) et de ChatGPT pour nous aider à faire le MakeFile ainsi que les fonctions rotations gauches et droites pour les AVL.

3) Limitations et difficultés

Notre programme est fidèle au cahier des charges néanmoins un problème persiste, la fonction de vérifications présente certains obstacles quant à son efficacité.

4) Exécution et programme

PRÉREQUIS

1. Télécharger tous les fichiers dans le même dossier.
2. Installer make et faire se donner les droits pour l'exécution de Home.sh.

```
sudo apt install make
```

```
chmod 777 Home.sh
```

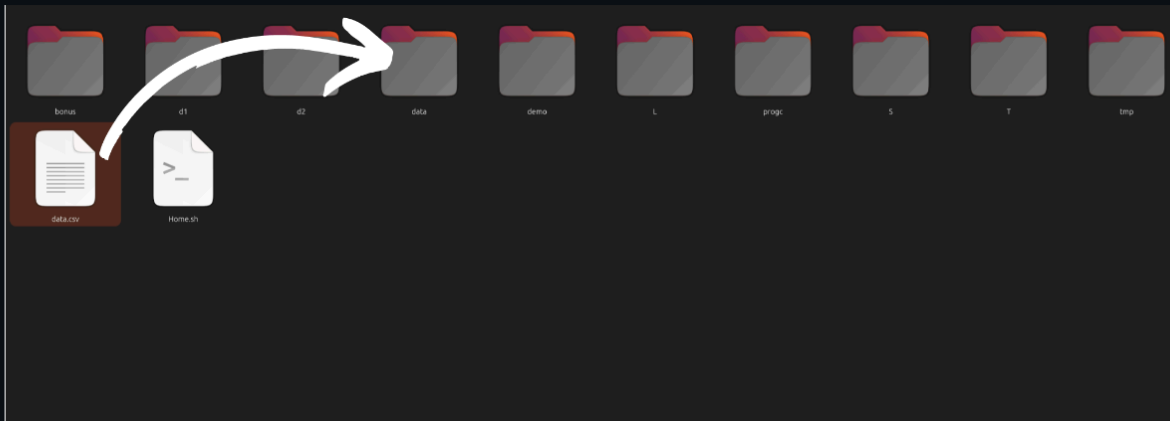
UTILISER AVEC UN INTERFACE

1. Entrer dans le dossier depuis le terminal

```
cd CY-TRUCKS
```

```
mkdir data
```

2. S'assurer de bien avoir mis le fichier source dans le dossier data



3. Exécuter le programme grâce à la commande ./Home.sh sans mettre AUCUN argument.

```
./Home.sh
```

UTILISER SANS INTERFACE

1. Récupérer le chemin d'accès du fichier

```
/chemindufichier
```

2. Entrer -h en argument pour accéder au menu

```
./Home.sh /chemindufichier -h
```

3. Exécuter le programme grâce à la commande ./Home.sh sans oublier les arguments.

```
./Home.sh /chemindufichier -d1 -d2 -l -s -t -bonus
```

MENU AVEC INTERFACE

CV-TAUCK

◆ Traitement ◆
Vérification de fichier
Quitter
Help
🖱️

- ▲ Pour monter dans l'interface
- ▼ Pour descendre dans l'interface
- Appuyer sur Entré pour validé
- Appuyer sur **H** pour aide

Dans le menu vous avez différents choix, vous pouvez choisir les traitements que vous souhaitez lancer et leur nombre

CV-TAUCK

◆ Conducteurs avec le plus de trajets ◆
Conducteurs et la plus grande distance
Les 10 trajets les plus longs
Les 10 villes les plus traversées
Statistiques sur les étapes
[Bonus] Conducteur avec le plus de ville traversé

- ▲ Pour monter dans l'interface
- ▼ Pour descendre dans l'interface
- Appuyer sur Entré pour validé
- Appuyer sur **H** pour Aide

Vous pouvez aussi vérifier que vos fichiers sont compatibles MAIS NE VOUS INQUIETEZ PAS ON LE VERIFIRA POUR VOUS. Vous pouvez aussi aller dans help si vous êtes perdu. Mais aussi vous pouvez arrêter le programme.

COMMANDS

Avec interface :

Utilisation :

- ▲ Pour monter dans l'interface
- ▼ Pour descendre dans l'interface
- Appuyer sur Entré pour validé

Liste des traitements disponibles :

Traitement -d1	Conducteur avec le plus de trajets	Temps d'execution 6 à 8 secondes
Traitement -d2	Conducteurs et la plus grande distance	Temps d'execution 3 à 5 secondes
Traitement -L	Les 10 trajets les plus longs	Temps d'execution 6 à 8 secondes
Traitement -T	Les 10 villes les plus traversées	Temps d'execution 14 à 16 secondes
Traitement -S	Statistiques sur les étapes	Temps d'execution 25 à 30 secondes
Traitement -Bonus	Conducteur avec le plus de trajet	Temps d'execution 26 à 30 secondes

Ajouter 5 sec en appuyant sur 'Entré'

Sans interface :

Utilisation :

Ecrire './Home.sh' avec différents arguments

Première argument : Chemin ou se trouve le fichier 'data.csv'
Arguments supplémentaires : Indiquer les traitements souhaité

- Pour executer traitement -d1 ajouter l'argument '-d1'
- Pour executer traitement -d2 ajouter l'argument '-d2'
- Pour executer traitement -L ajouter l'argument '-L'
- Pour executer traitement -T ajouter l'argument '-T'
- Pour executer traitement -S ajouter l'argument '-S'
- Pour executer traitement -bonus ajouter l'argument '-bonus'

Vous pouvez utiliser les flèches pour vous déplacer dans les différents menus sans oublier que 'h' permet de d'ouvrir l'option aide

AIDE



Utilisation :

- ▲ Pour monter dans l'interface
- ▼ Pour descendre dans l'interface
- Appuyer sur Entrée pour valider

Liste des traitements disponibles :

Traitement -d1	Conducteur avec le plus de trajets	Temps d'exécution 6 à 8 secondes
Traitement -d2	Conducteurs et la plus grande distance	Temps d'exécution 3 à 5 secondes
Traitement -L	Les 10 trajets les plus longs	Temps d'exécution 6 à 8 secondes
Traitement -T	Les 10 villes les plus traversées	Temps d'exécution 14 à 16 secondes
Traitement -S	Statistiques sur les étapes	Temps d'exécution 25 à 30 secondes
Traitement -Bonus	Conducteur avec le plus de trajet	Temps d'exécution 26 à 30 secondes

Ajouter 5 sec en appuyant sur 'h'

NOMBRE DE TRAITEMENT A SPECIFIER

ici vous pouvez spécifier votre nombre de traitement avec une aide

Liste des traitements disponibles :

Traitement -d1	Conducteur avec le plus de trajets	Temps d'exécution 6 à 8 secondes
Traitement -d2	Conducteurs et la plus grande distance	Temps d'exécution 3 à 5 secondes
Traitement -L	Les 10 trajets les plus longs	Temps d'exécution 6 à 8 secondes
Traitement -T	Les 10 villes les plus traversées	Temps d'exécution 14 à 16 secondes
Traitement -S	Statistiques sur les étapes	Temps d'exécution 25 à 30 secondes
Traitement -Bonus	Conducteur avec le plus de trajet	Temps d'exécution 26 à 30 secondes

Entrez le nombre de traitement que vous souhaitez faire (entre 1 et 6) :