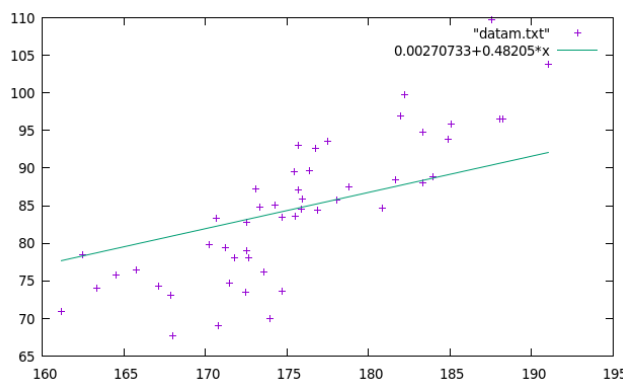


Dans ce TP, nous avons du utilisé une base de donnée et créer une sorte de reseau de neurones qui, à partir des ces données d'entrée permet de prédire le poid en fonction de la taille et du sexe de l'individu.

Pour ce faire, nous avons commencé par créer un fichier CmakeLists contenant toute les instruction pour aboutir à un make sans avoir à faire le makefile nous même. Il est indispensable de passer par un makefile dans un projet de plusieurs fichiers. Dans notre cas, il aurait été possible de compiler chaque exercice à la main individuellement mais il est plus pratique et moins source d'erreur de par par un makefile ou cmake quand même.

Le TP s'est divisé en 3 exercice similaires dans le traitement, avec une base de donnée en entrée que l'on traite, analyse et examine avant de ressortir la prédiction demandée par l'utilisateur.

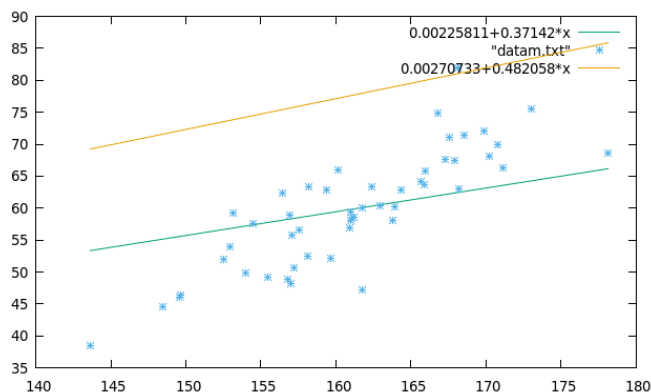
### EXO 1 HOMME :



Dans cette première partie, nous nous somme intéressé à un jeu de donnée concernant des hommes. A l'aide de plusieurs fonction, nous analysons les donnée, trouvons une équation qui prend en argument la taille pour prédire le poid de l'homme. Ceci s'apparente a un simple reseau de neurones.

On peut voir ici a gauche la courbe de tendance défini lors de l'exécution du programme sous forme de graphique à l'aide de gnuplot.

### EXO 1 FEMME :



Tout simplement, nous avons effectué la même tâche avec des données sur des femmes.

Le graphique à gauche nous montre la courbe de tendance des femmes en blue et en orange, la courbe des homme d'au dessus pour comparer. On constate sans grande surprise un poid plus faible chez les femmes par rapport aux hommes de même taille.

Question 12 : comment faire pour factoriser du code entre les fichiers ex1m.cpp et ex1f.cpp :

en réalité, les ex1m et ex1f n'ont qu'une seul chose en commun, la data d'entrée, on pourrais simplement écrire un seul code qui prédit le poid d'une personne selon sa taille et son sexe, c'est exactement ce que nous avons fait dans l'exercice 2.

On pourrais aussi rendre le main propre et mettre toutes les fonctions ailleurs que dans le fichier du main et passer par un fonction.cpp et son header fonction.hpp par exemple.

## EXO 2 HOMME/FEMME :

Dans ce 2ème exercice nous mettons justement en pratique la fusion des data et prendre en argument le sexe de la personne aussi. Rien de bien compliqué a cela quand on à déjà les fonctions qui prédisent le poids sans le sexe. Il suffit donc de rajouter quelques lignes de code pour bien prendre en compte quel est le sexe de la personne.

Nous avons aussi généré un fichier cost.txt qui nous montre une convergence du coups de calcul de notre prédiction en fonction du nombre d'itération. On se rend assez rapidement compte du la vitesse à laquelle la convergence se produit lorsqu'en seulement 50 itérations, nous passons de 2050 à 25.

Question 9 : comment faire pour factoriser du code entre les fichiers ex1m.cpp, ex1f.cpp et ex2.cpp :

Cet exercice avais justement pour but de factoriser du code entre les ex1m et ex1f afin de créer un code plus général et unique. Encore une fois, nous aurions pu passer par un fichier de fonctions et son header pour liberer le main de toute fonctions et de le laisser le plus clair et bref possible.