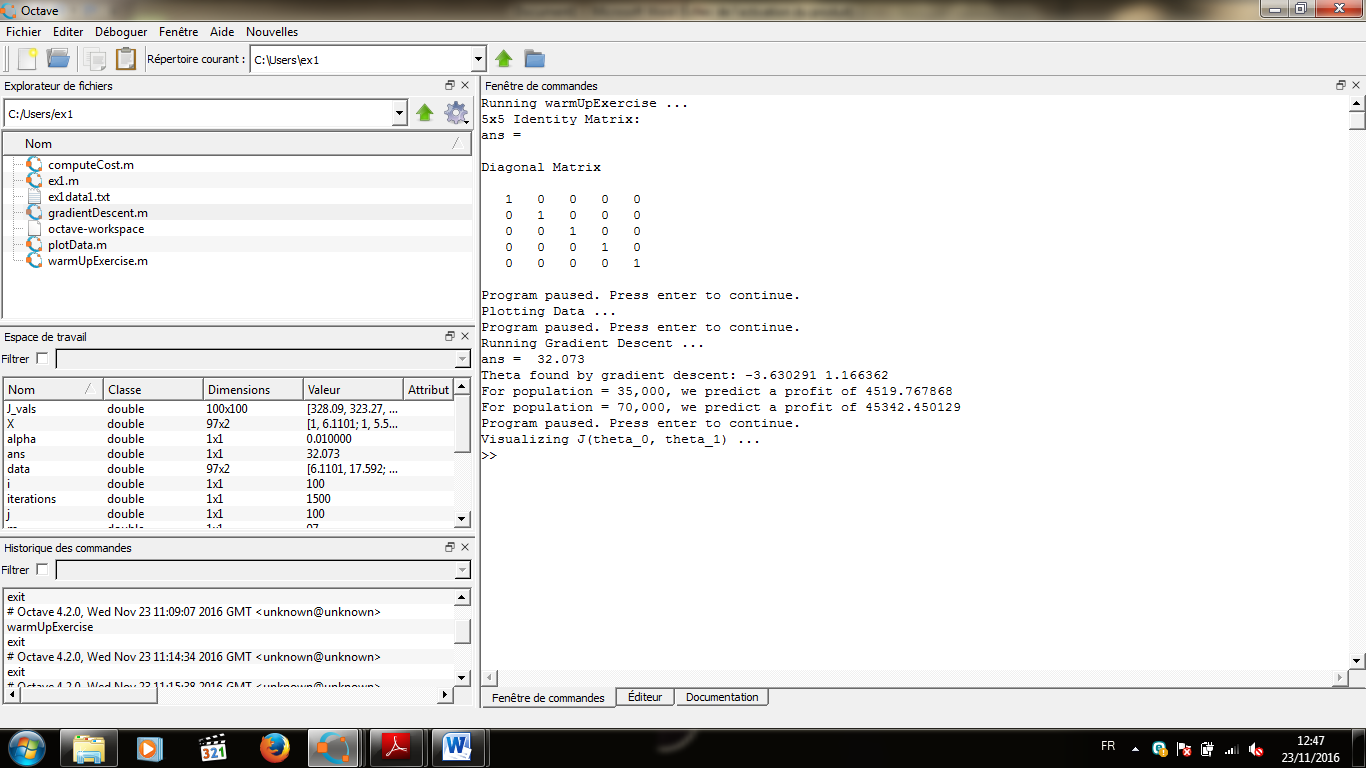
# TP1 Data Mining : Régression Linéaire

## Warm up :

Dans cette partie nous allons voir comment Octave fonctionne, et comment créer des fonctions avec cet outil de simulation numérique. Le script warmUpExercice nous aide à comprendre comment on crée une fonction sous Octave. Le but est ici de créer une fonction qui nous renvoie la matrice identité de taille 5x5. Cette fonction est reprise dans le script principal et nous permet de voir la matrice souhaitée :



## Plot Data

Le but de ce TP est de modifier les différentes fonctions pour arriver à la régression linéaire des points donnés dans un fichier texte modélisant en abscisse la population d’une ville et en ordonnée les profits réalisés par un camion de nourriture dans cette ville. Dans cette partie nous allons afficher dans un graphe en 2D les données qui concernent notre problème. Nous travaillerons sur la fonction plotData(x,y) qui prend en argument x(l’abscisse) y (l’ordonnée) le type de marqueur et la taille du marqueur. La figure va ensuite être créée dans une nouvelle fenêtre.

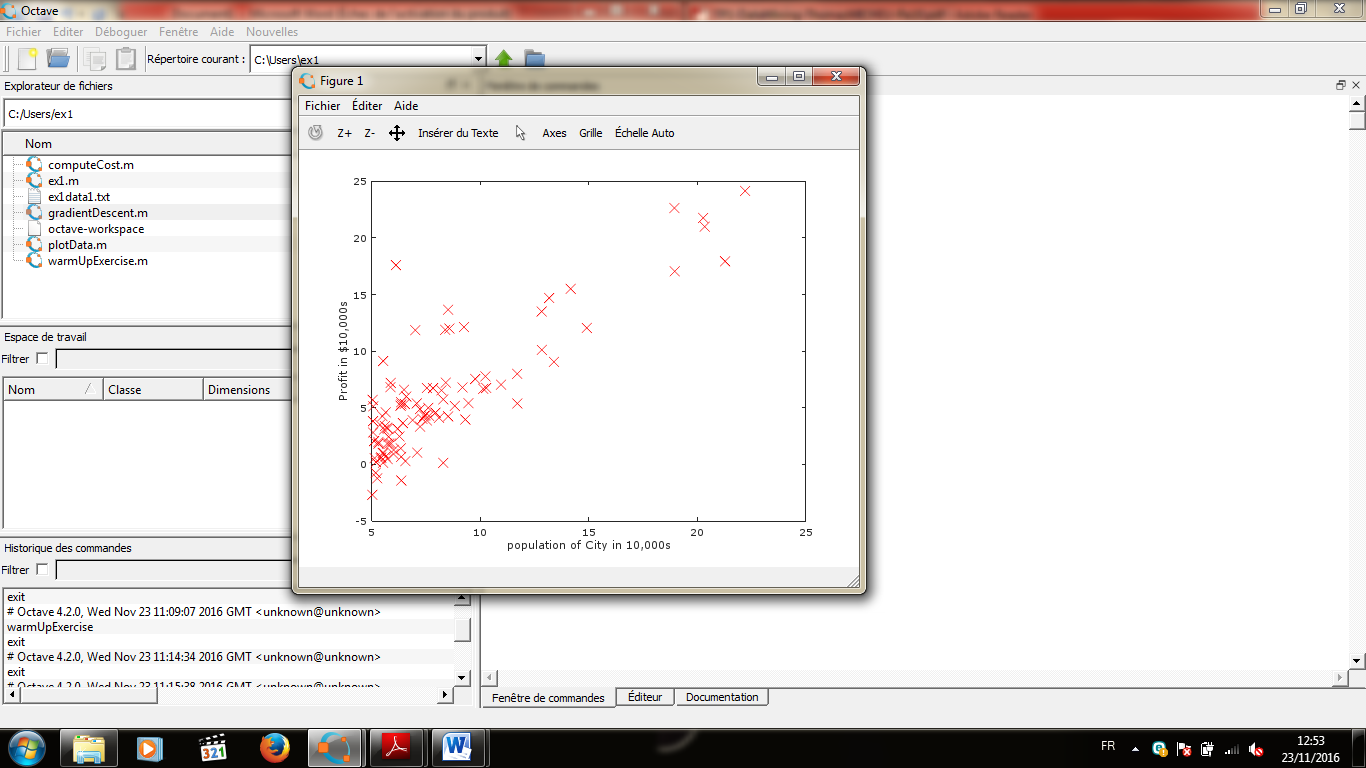


Figure - Fonction PlotData

Sur cette figure il y a quelques disparités mais on observe une tendance linéaire de la relation entre la population d’une ville et les profits générés.

## Régression linéaire

Dans cette partie on cherche à établir la relation de linéarité entre la population d’une ville et les profits réalisé par le camion passant dans cette ville. Pour ce faire, on utilise la loi vue en cours :

La précision de cette loi est donnée par la fonction suivante :

On mesure la distance au carrée moyenne entre les y(i) et les valeurs obtenues par hϴ. Pour cela on utilise la méthode du gradient afin de déterminer ϴ0 et ϴ1 qui vont minimiser la fonction hϴ. Cet algorithme du gradient est itératif et permet d’effectuer une correction des termes ϴj à chaque tour de boucle avec la formule suivante :

Les termes ϴ0 et ϴ1 vont donc converger vers des valeurs qui minimisent la fonction J(ϴ)

La fonction computeCost() calcule la valeur de 𝐽(𝜃) à partir des arguments d’entrée X : nombres d’habitants des villes, y : profits générés dans chaque ville, et 𝜃, vecteur contenant les coefficients ϴ 0 et ϴ 1.

Les valeurs finales de ϴ 0 et ϴ 1 sont affichées dans la fenêtre de commande. On obtient ϴ0=−3.63 et ϴ1=1.17.

Avec une ligne de commande supplémentaire, on ajoute sur la figure 1 du PlotData() la courbe de régression linéaire en bleue.

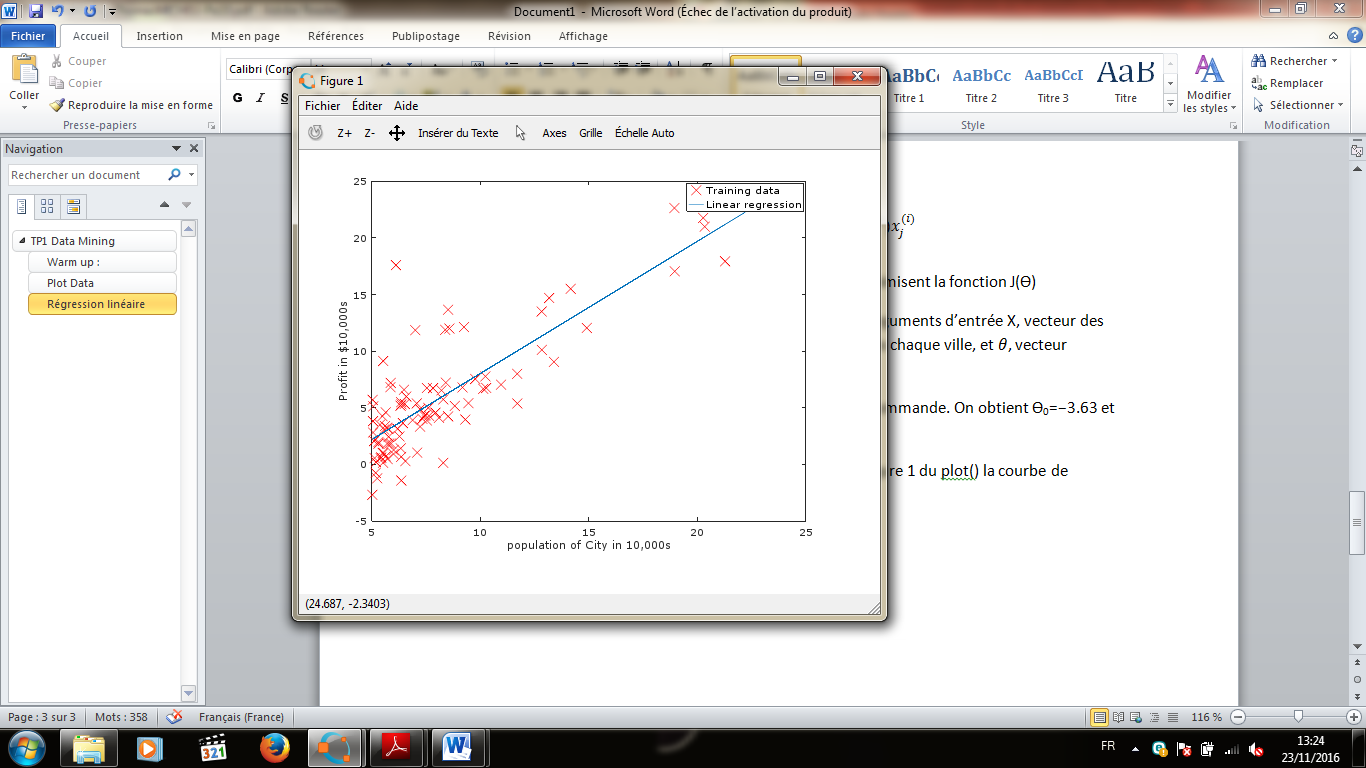
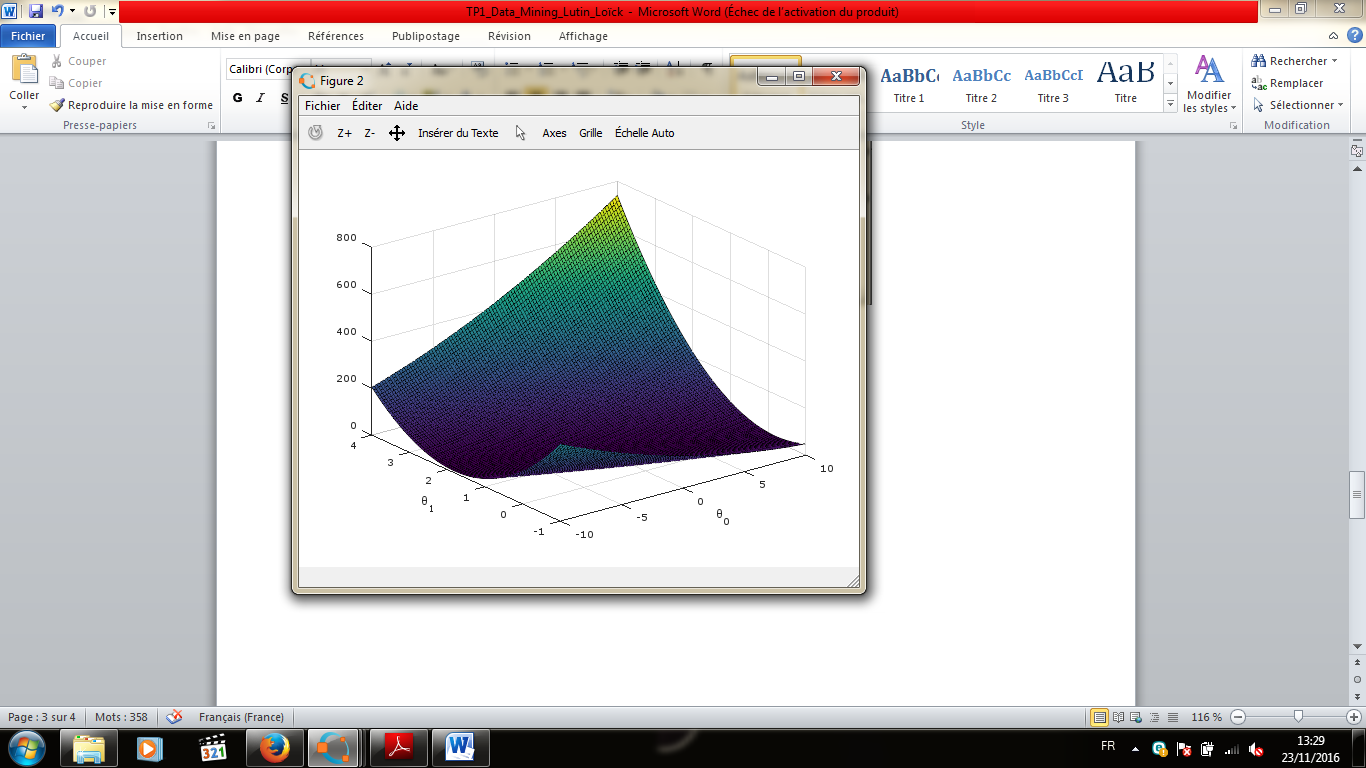


Figure - régression linéaire

## Graphes fonction coût et gradient

On crée une grille de theta pour laquelle on va pouvoir vérifier que l’on a bien minimiser les valeurs de ϴ 0 et ϴ 1. Les valeurs de ϴ sont prises dans [(-10 ; 10) x (-1 ; 4)], avec 100 valeurs sur les deux axes.

On crée une matrice de coût dans laquelle on rentre les valeurs de J(ϴ) avec les numéros de ligne et de colonnes correspondant aux deux theta



On trace un second graphe en 2D qui représente l’état de convergence des variables ϴ 0 et ϴ 1

