

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**Université Benyoucef BENKHEDDA- Alger1**

Faculté des Sciences

Département **Mathématiques et Informatique**



# **RAPPORT TP**

**Informatique**

*Spécialité : ISII*

---

## **Régression Linéaire**

---

**Réalisé par**

- Benbaba Rym Amina

**Groupe**

02

2018/2019

## Avant de commencer ....

Ce TP a été implémenter dans octave et il inclut les fonctions suivantes :

- ✓ lab\_Benbaba.m
- ✓ hypothesis\_Benbaba.m
- ✓ Cost\_Benbaba.m
- ✓ GradientDescent\_Benbaba.m
- ✓ Plot\_Hypothesis\_Benbaba.m
- ✓ plot\_Cost\_Benbaba.m

Tout en utilisant les fichiers Input1.txt et Output.txt

## Les fonctions :

### 1. hypothesis\_Benbaba.m

```
hypothesis_Benbaba.m ✖
1  ## Copyright (C) 2018 Administrateur
2
3
4  ## Author: Administrateur <Administrateur@RYM-PC>
5  ## Created: 2018-12-21
6
7  function hypo=hypothesis_Benbaba(theta, X)
8      hypo=X*theta
9
10 endfunction
11
```

### 2. Cost\_Benbaba.m

```
Cost_Benbaba.m ✖
1
2  ## Author: Administrateur <Administrateur@RYM-PC>
3  ## Created: 2018-12-21
4
5  function cost = Cost_Benbaba (X, y ,theta)
6      #initialisation
7      #m=taile de vecteur y
8      m=length(y);
9
10     h=hypothesis_Benbaba(theta, X); #calculé d'hypothèse
11     cost=(1/(2*m))*(sum((h-y).^2)); #la fonction cost
12
13 endfunction
14
```

### 3. GradientDescent\_Benbaba.m

```
GradientDescent_Benbaba.m
1
2  ## Author: Administrateur <Administrateur@RYM-PC>
3  ## Created: 2018-12-21
4
5  function [theta, cost_vect] = GradientDescent_Benbaba (X, y, theta, alpha, iterations)
6  %initialisations
7  m=length(y); #taille de vector y
8  cost_vect=zeros (iterations, 1); #tab nbrligne=iterations
9  %boucler de 1 à itérations
10 for i=1 : iterations,
11     #h=x*theta;
12     h=hypothesis_Benbaba(theta, X);#recuperer la valeur de hypothèse
13     t1=theta(1) - (alpha * (1/m) * sum(h-y));
14     t2=theta(2) - (alpha * (1/m)*sum((h-y).*X(:,2)));
15     %update theta(1) et theta(2)
16     theta(1) =t1;
17     theta (2)=t2;
18     cost_vect(i)= Cost_Benbaba(X,y,theta);
19 endfor
20
21 endfunction
22
```

### 4. Plot\_Hypothesis\_Benbaba.m

```
Plot_Hypothesis_Benbaba.m
1  ## Copyright (C) 2018 Administrateur
2  ##
3
4
5  ## Author: Administrateur <Administrateur@RYM-PC>
6  ## Created: 2018-12-21
7
8  function Plot_Hypothesis_Benbaba (theta, X)
9
10     plot(X(:, 2), X * theta, '-');
11 end
```

### 5. plot\_Cost\_Benbaba.m

```
plot_Cost_Benbaba.m ✖
1  ## Copyright (C) 2018 Administrateur
2  ##
3
4
5  ## Author: Administrateur <Administrateur@RYM-PC>
6  ## Created: 2018-12-21
7
8  function plot_Cost_Benbaba (iterations, cost_vect)
9      plot(1: iterations, cost_vect, '-r');
10
11 endfunction
12
```

## 6. lab\_Benbaba.m

```
lab_Benbaba.m ✖
1
2  ## Author: Administrateur <Administrateur@RYM-PC>
3  ## Created: 2018-12-21
4  ## clear : supprime les variables de Workspace
5  ## close all : ferme toutes les figures ouvertes
6  ## clc :pour supprimer tout ce qui affiché dans cmd
7
8  clear; close all; clc;
9  graphics_toolkit('gnuplot') #pour les grandes nombres
10 X = load('input1.txt');
11 y = load('Output.txt');
12 m = length(y); #=97 dans ce cas taille de vector y
13 theta = zeros(2, 1); #tab de 2ligne et une colonne initialiser à 0
14 iterations = 50;#nbr ditérations'
15 alpha = 0.025;
16 %plot hypothesis
17 plot(X, y, 'bx');
18 title('GRAPHE'); #titre de graphe
19 xlabel('Surface maison');#nom dans ave des X
20 ylabel('Prix');#nom dans axe des y
21
22 %add 1 vecteur de 1 dans X0 au vecteur X
23 X = [ones(size(X,1), 1), X];
24 %la fonction cost
25 cost = Cost_Benbaba(X, y, theta); #calculer l'erreur dans cost
```

```

%add 1 vecteur de 1 dans X0 au vecteur X
X = [ones(size(X,1), 1), X];
%la fonction cost
cost = Cost_Benbaba(X, y, theta); #calculer l'erreur dans cost
%fonction gradient descent
[ theta, cost_vect] = GradientDescent_Benbaba(X, y, theta, alpha, iterations);
#pour calculer theta et nvl cost_vect de la fonction cost

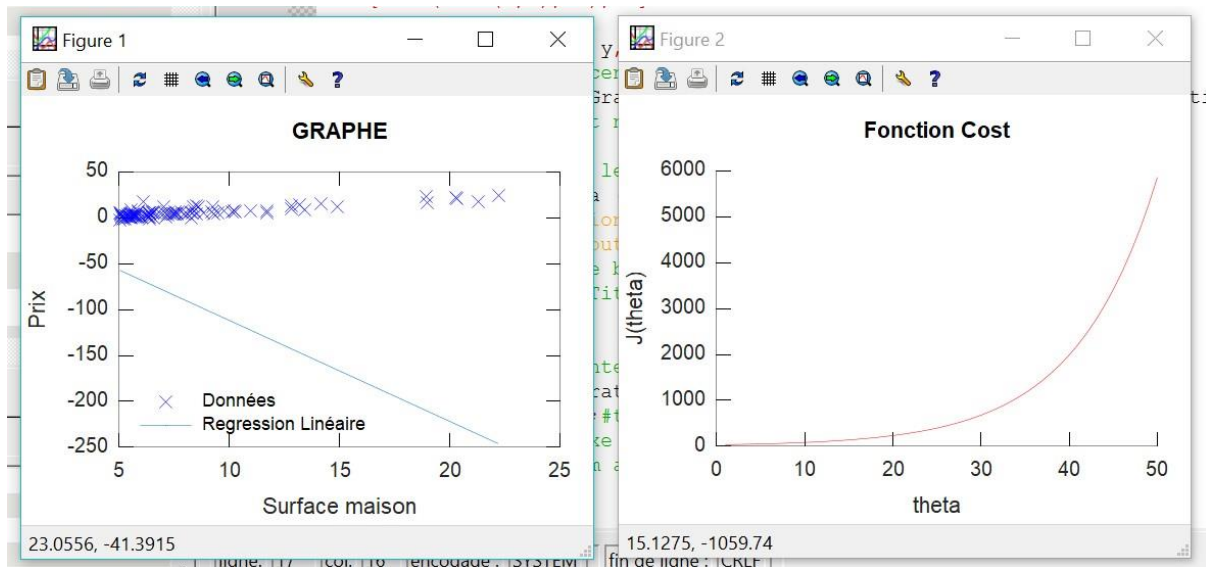
hold on;#affichage sur le même graphe
Plot_Hypothesis_Benbaba (theta, X)#regression linéaire
l={'Données', 'Regression Linéaire'};
legend(l,'Location','southwest'); #position de legend
legend('boxoff')#pas de box
#title(lgd,'My Legend Title')
hold off;
#crée autre figure
figure(2);hold on;#maintenant on affiche dans un autre graphe
plot_Cost_Benbaba (iterations, cost_vect)
title('Fonction Cost');#titre de graphe
xlabel('theta');#nom axe x
ylabel('J(theta)');#nom axe y

hold off;

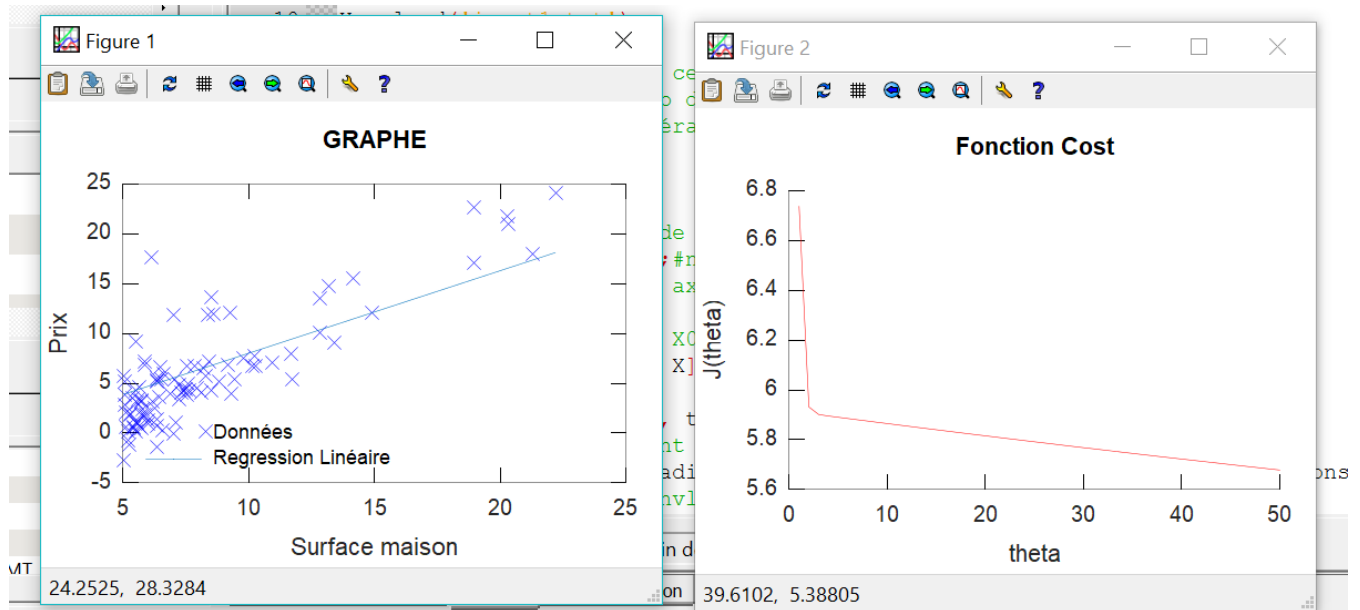
```

Et voici les deux graphes qui s'affichent lors de l'exécution de code

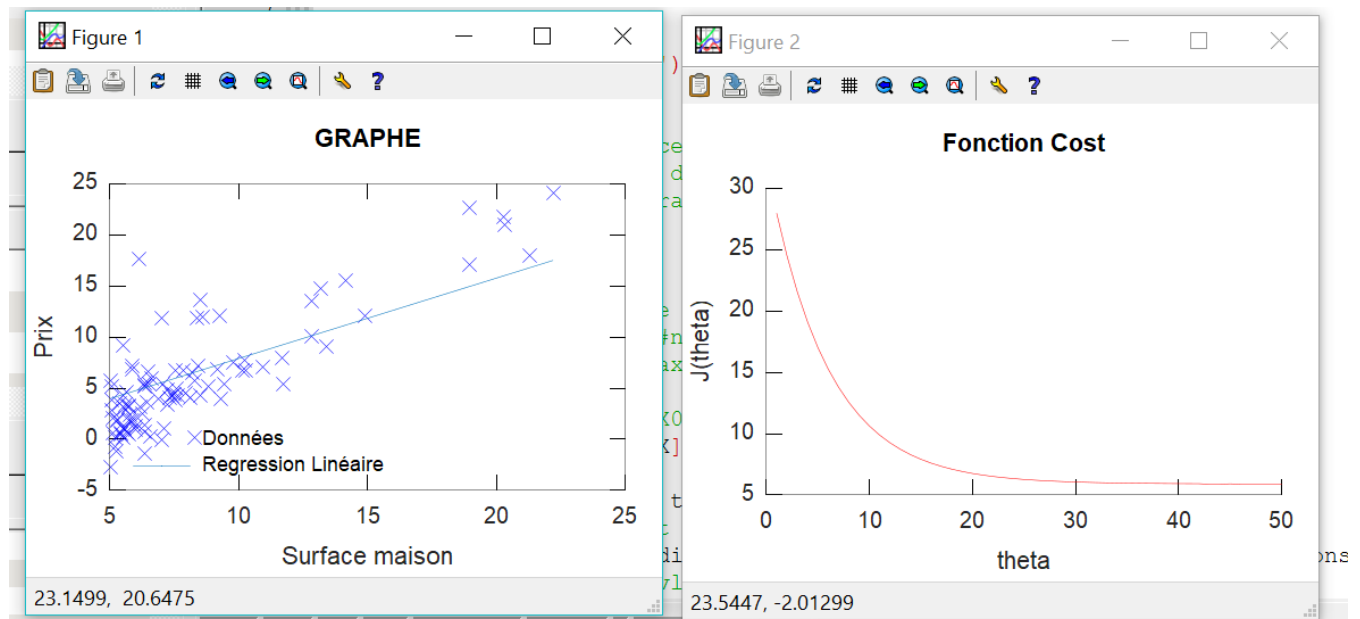
(lab\_Benbaba.m) :



Pour  $\alpha=0,01$  on à :



Pour  $\alpha = 0,001$



### **Conclusion :**

On remarque que pour  $\alpha = 0,025$  l'erreur augmente et la régression linéaire diverge. Il faut utiliser une valeur petite pour  $\alpha$  (0,01 ---- 0,001) pour que le GradientDescent fonctionne, même pour la fonction cost.