Lab 02: elements of corrections

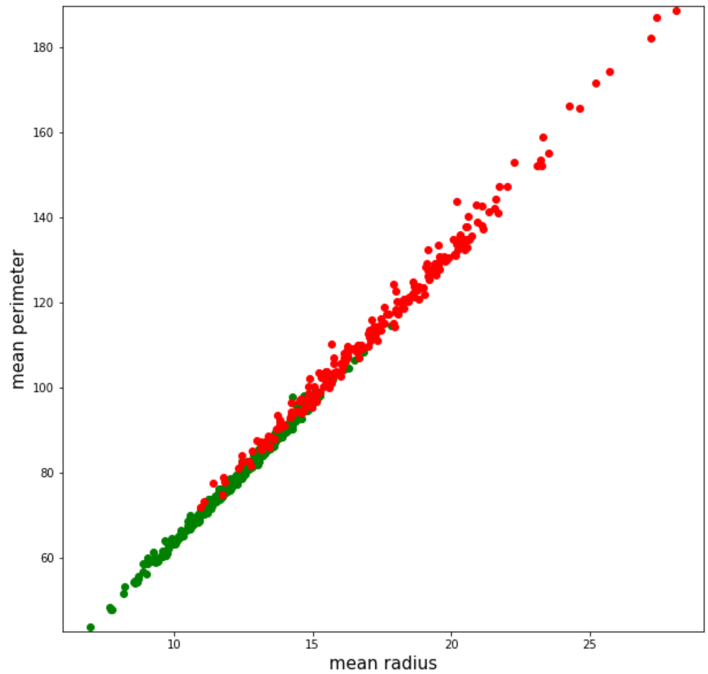
Q1 : 569 samples and 31 potential features

Q2 : 31 potential features:

- mean radius
- mean texture
- mean perimeter
- mean area
- mean smoothness
- mean compactness
- mean concavity
- mean concave points
- mean symmetry
- mean fractal dimension
- radius error
- texture error
- perimeter error
- area error
- smoothness error
- compactness error
- concavity error
- concave points error
- symmetry error
- fractal dimension error
- worst radius
- worst texture
- worst perimeter
- worst area
- worst smoothness
- worst compactness
- worst concavity
- worst concave points
- worst symmetry
- worst fractal dimension
- label

Q3 : 357 benign cells (63.0% of the dataset)
212 malignant cells (37.0% of the dataset)

Q4 :



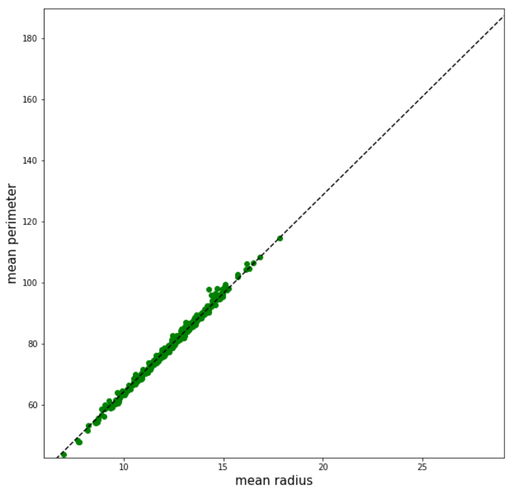
Q5 : easy!

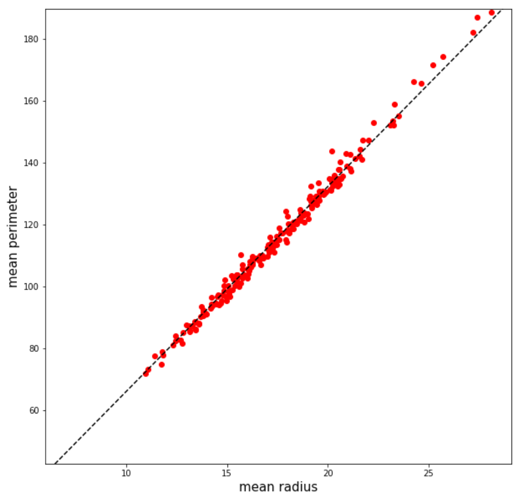
Q6 : Implementation using the mean function from Numpy, the operator **2, the variables p1, p2, lr and alpha :

- $$\frac{\partial L}{\partial \alpha_k} = \frac{-2}{n} \sum_{i=1}^{n} r_i (p_i - \alpha_k * r_i)$$
- $$\alpha_{k+1} = \alpha_k - lr * \frac{\partial L}{\partial \alpha_k}$$
- $$L(\alpha_{k+1}) = \frac{1}{n} \sum_{i=1}^{n} (p_i - \alpha_{k+1} * r_i)^2$$

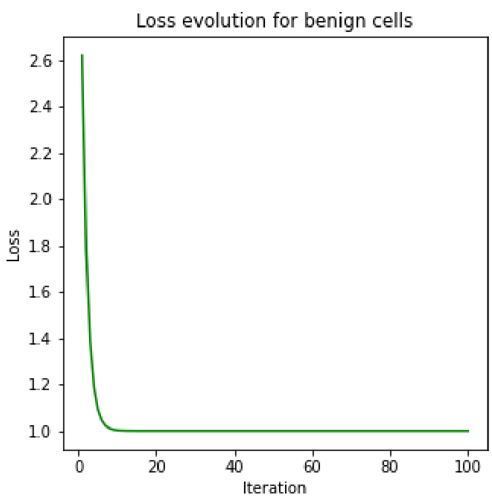
Q7 :

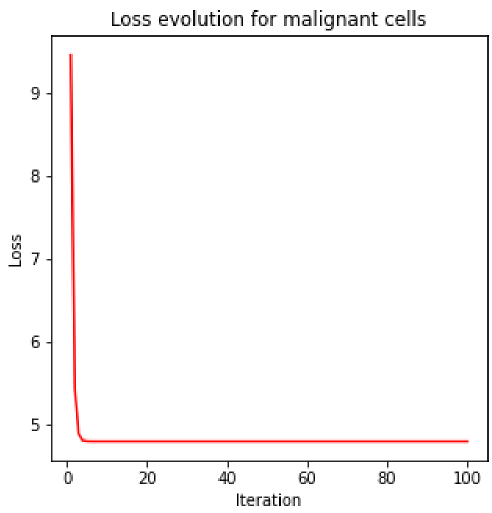
```
Linear regression on benign cells:
Initial alpha = 6.283185307179586 & loss = 4.320215995692195
Optimized Alpha = 6.43162179471489 & loss = 0.9997976094672045

Linear regression on malignant cells:
Initial alpha = 6.283185307179586 & loss = 38.93427064426372
Optimized Alpha = 6.612258213059991 & loss = 4.805110085338104
```





Q9 :

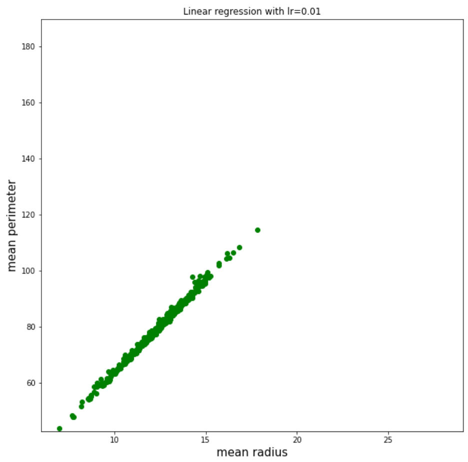


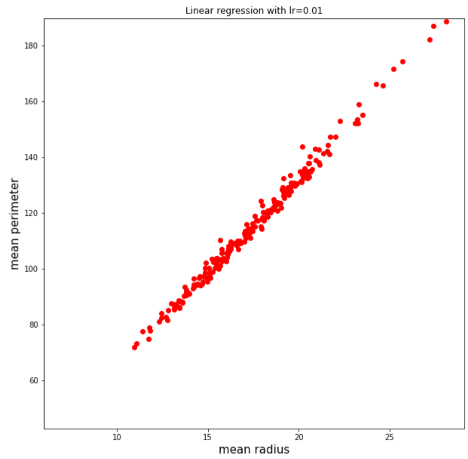


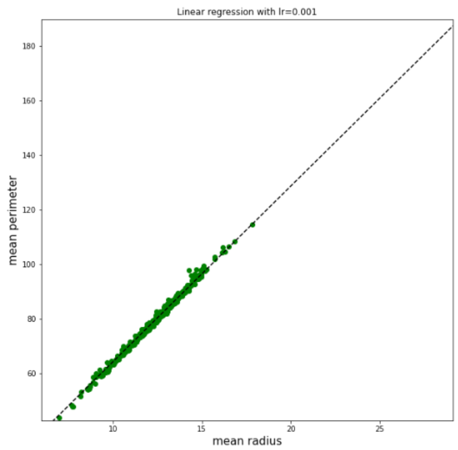
Q10 :

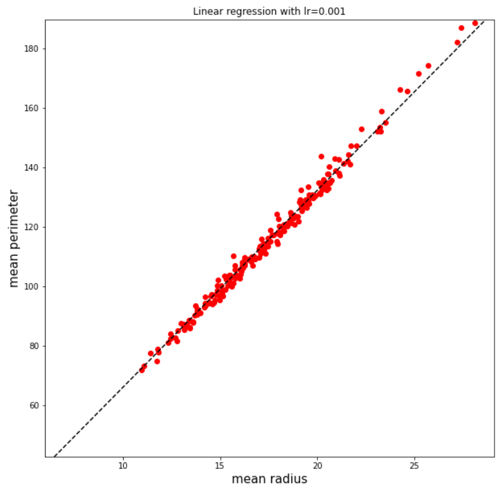
```
Learning rate = 0.01
Linear regression on benign cells:
Initial alpha = 6.283185307179586 & loss = 4.320215995692195
Optimized Alpha = -3.777642449040333e+29 & loss = 2.1505679623477766e+61

Linear regression on malignant cells:
Initial alpha = 6.283185307179586 & loss = 38.93427064426372
Optimized Alpha = -9.382367737162349e+71 & loss = 2.7743819849433777e+146

Learning rate = 0.001
Linear regression on benign cells:
Initial alpha = 6.283185307179586 & loss = 4.320215995692195
Optimized Alpha = 6.43162179471489 & loss = 0.9997976094672045

Linear regression on malignant cells:
Initial alpha = 6.283185307179586 & loss = 38.93427064426372
Optimized Alpha = 6.612258213059991 & loss = 4.805110085338104
```

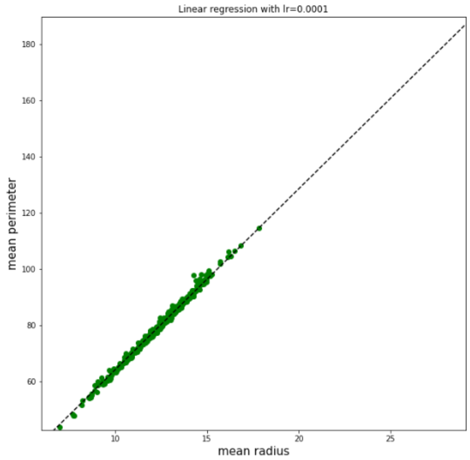
```
Learning rate = 0.0001
Linear regression on benign cells:
Initial alpha = 6.283185307179586 & loss = 4.320215995692195
Optimized Alpha = 6.4246643300655455 & loss = 1.0070924111340778

Linear regression on malignant cells:
Initial alpha = 6.283185307179586 & loss = 38.93427064426372
Optimized Alpha = 6.61176877361184 & loss = 4.805185584006858
```

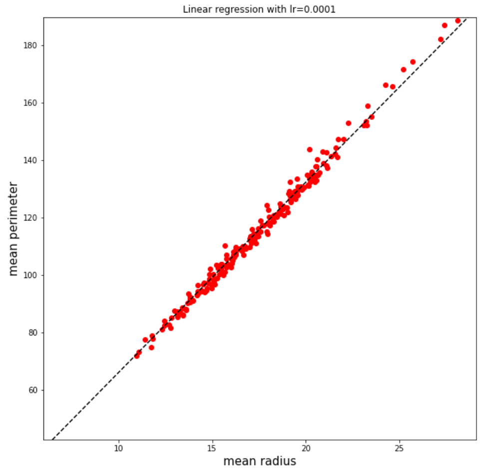




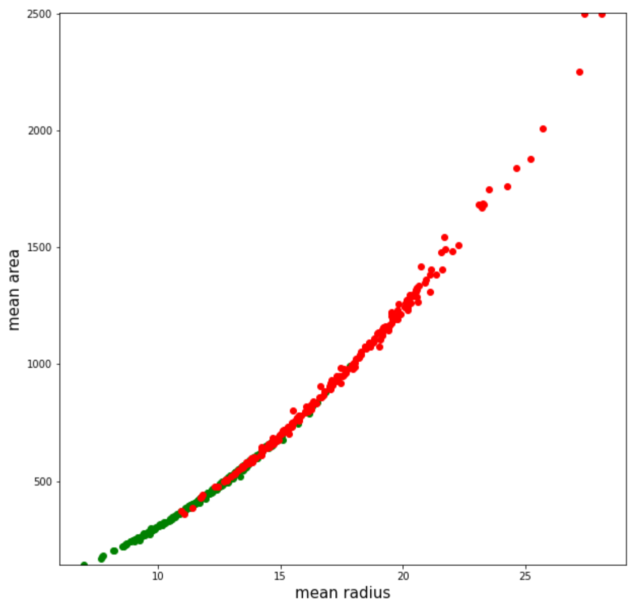








Q11 :



Q12: easy!

Q13: implement $L(\alpha, \beta, \gamma) = \frac{1}{n}\sum_{i=1}^{n}(a_i - \alpha * r_i^2 - \beta * r_i - \gamma)^2$ using the mean function from Numpy, the operator **2, the variables p1, p2, alpha, beta and gamma

Q13bis:

$$\frac{\partial L}{\partial \alpha} = \frac{-2}{n}\sum_{i=1}^{n} r_i^2(a_i - \alpha r_i^2 - \beta r_i - \gamma)$$

$$\frac{\partial L}{\partial \beta} = \frac{-2}{n}\sum_{i=1}^{n} r_i(a_i - \alpha r_i^2 - \beta r_i - \gamma)$$

$$\frac{\partial L}{\partial \gamma} = \frac{-2}{n}\sum_{i=1}^{n}(a_i - \alpha r_i^2 - \beta r_i - \gamma)$$

Q14: you have to apply the conjugate gradient algorithm (first order method). By default, Scipy uses BFGS (second order method).

```
Initial parameters: alpha = 3.141592653589793, beta=0, gamma=0

Quadratic regression on benign cells:
Initial loss: 140.1446267318879
Optimized parameters: alpha=2.990274955746421, beta=2.8015718254897592,
gamma=-21.871751262738098
Optimized loss: 24.065946376409975

Quadratic regression on malignant cells:
Initial loss: 684.1691819607581
Optimized parameters: alpha=3.1217338123002145, beta=-0.30251734131210206,
gamma=-0.209652311544625555
Optimized loss: 535.1980389810111
```

Results' visualization (not asked):