Generate random data from a predefined linear regression model. In particular, consider that $0 \leq \beta_i \leq 5 \quad i = 0, \ldots, K$ are integer with at least $K = 1000$ independent variables $X = (X_1, X_2, \ldots, X_K)$ and $n = 50000$ observations. We seek to adjust a multiple linear regression model to explain variable Y as a function of the other variables X, i.e., ($Y = \boldsymbol{\beta}' X + \epsilon$) by using "Ridge Regression" under a least-squares approach:

$$\min_{\beta} \quad ||y - X\beta||_2^2 + \rho||\beta||_2^2 \tag{1}$$

where ρ is a parameter of your election (for instance consider $\rho = 5$).

a) (*0.5 points*) Estimate the value of the regression coefficients by implementing the analytical solution. Use this solution as a benchmark for the following sections.

b) (*1 points*) Estimate the value of the regression coefficients by using the function **minimize** from the Python module **Scipy.optimize**. Try at least four available solvers and compare their performance in terms of number of iterations, number of function, gradient and hessian evaluations as well as total computational time.

c) (*1 points*) Modify the preceding optimization model by adding (lower and upper) bounds on the values of the β coefficients. Solve it again with the module **Scipy.optimize** a by using (at least) two different solvers, which should accept the introduction of bounds on the variables. Compare these methods and briefly comment on possible interpretations of the values of the coefficients.

d) Estimate the value of the regression coefficients of (1) by implementing the:*****

   i. (*0.5 points*) **Gradient** method.

   ii. (*0.5 points*) **Newton** method.

   iii. (*1 points*) **Quasi-Newton** method.

   Consider a line search technique to improve the algorithm convergence, e.x., Armijo rule. Compare the performance of these algorithms (number of iterations, function, gradient and hessian evaluations, and total computational time).

e) Estimate the value of the regression coefficients of (1) by implementing the:*****

   i. (*0.5 points*) **Coordinate descent** method.

   ii. (*0.5 points*) **Stochastic gradient** method.

   iii. (*2.5 points*) Three other techniques presented in Topic 2 (one from each category: "noise reduction", "second-order" and "other" methods).

   Compare the performance of these algorithms (number of iterations, function, gradient and hessian evaluations, and total computational time).

f) (*2 points*) Consider the constrained problem:

$$\min_{\beta} \quad ||y - X\beta||_2^2 \tag{2a}$$

$$\text{s.t.} \quad \sum_{i=1}^{K} \beta_i \leq 100 \tag{2b}$$

Estimate optimal value of the regression coefficients in (2) by implementing a barrier algorithm.*

*You have to write down the algorithms' code in `Python` by yourself.

IMPORTANT:

- Due date: **Monday, May 9th, at 11 p.m.**
- Upload the code to Aula Global as a Jupyter notebook.
- Name your notebook file as "Surname-Name-A1.ipynb".
- You are strongly advised to include descriptions for your formulations and comments in the same notebook, by using markdown cells.
- If this would prove too complicated, exceptionally you may present this information in a separate pdf file. In this case, name the file "Surname-Name-A1.pdf".
- Upload any datasets that might be required to reproduce your results (if not included with the models).