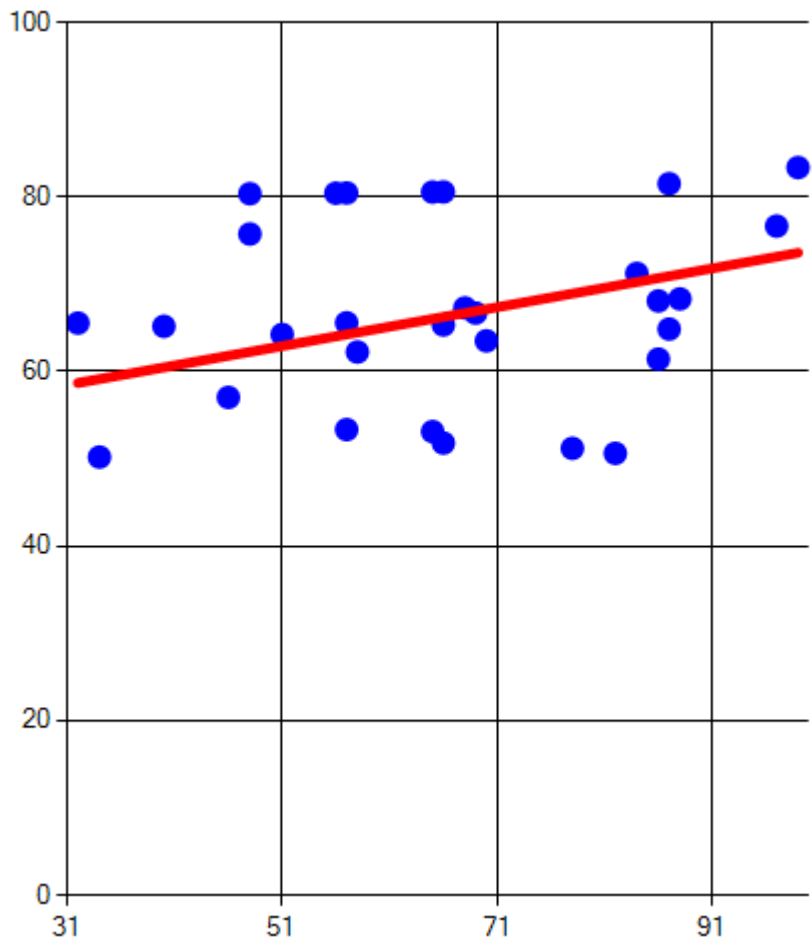


Задача

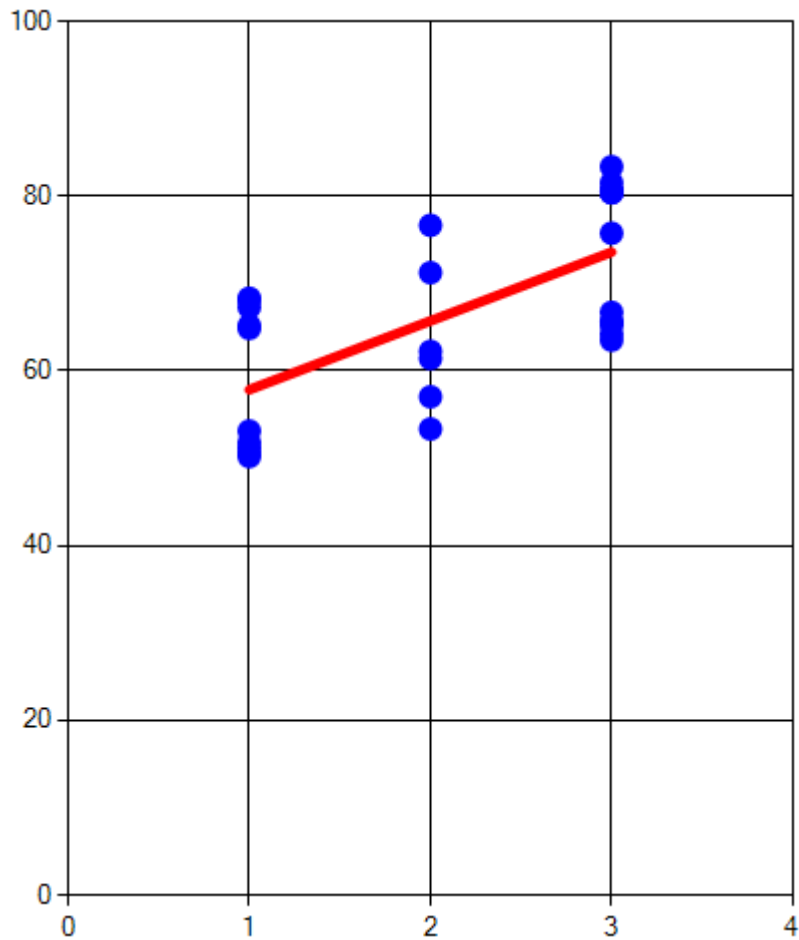
- Банка разполага с капитал $W = 300000$ лв., който може да предостави на кредитополучатели. Всеки клиент заявява сума w_i (за еднакъв период). Банката очаква клиентът да възстанови сумата плюс лихва p_i .
- Известни данни за предишни клиенти (csv):
 - Години (18-...);
 - Образование (1-основно, 2-средно, 3-висше);
 - Взета сума;
 - Изпълнение.
- Да се състави алгоритъм, разпределящ капитала по клиенти (csv), така че да реализира максимална печалба.
 - Години (18-...);
 - Образование (1-основно, 2-средно, 3-висше);
 - Заявена сума.
- CSVs: <https://ufile.io/8p2c23jq>

Изпълнение спрямо:

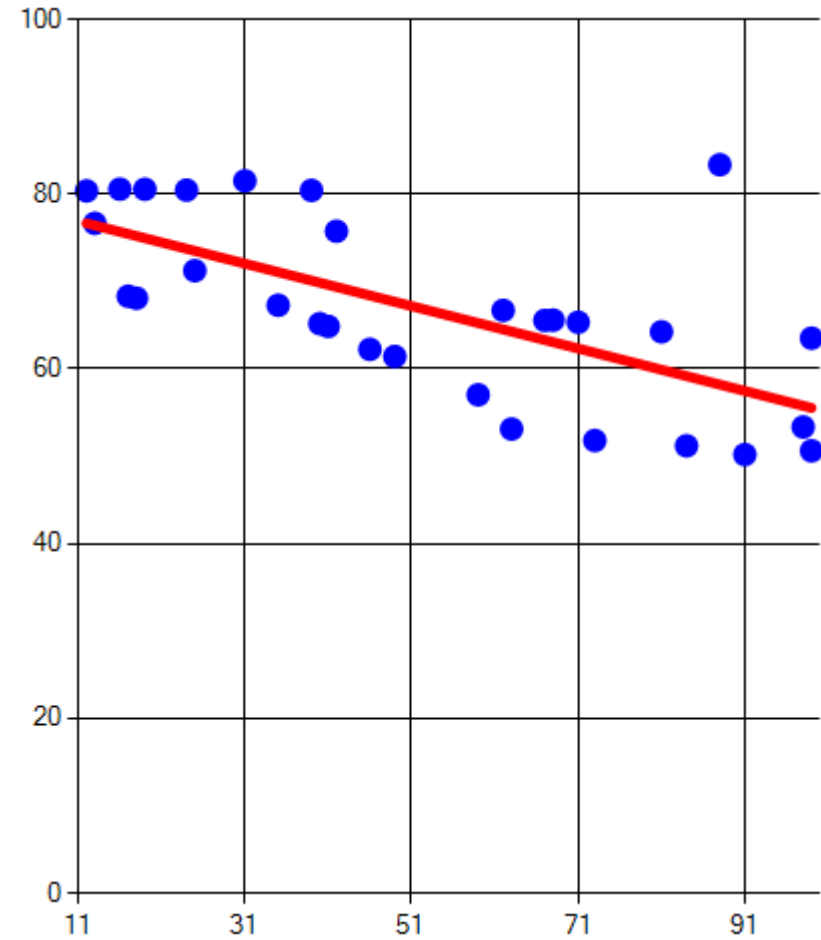
- Възраст



- Образование



- Заявена сума



Задача

- Зареждане на данните от data.csv:

- x_1 - години;
- x_2 - образование;
- x_3 - заявена сума;
- y - изпълнение.

- Обучаване на линеен модел по данните (лекция 11):

$$\theta_0 + \theta_1 x_1 + \theta_2 x_x + \theta_3 x_3 = \hat{y}$$

- Решаване на задача за раницата по отношение на данните в request.csv, като w_i е заявената сума, p_i е заявената сума * очаквано изпълнението.

```
var data = File.ReadAllLines("data.csv")
    .Skip(1)
    .Select(s =>
        s.Split(new [] { ';' }, StringSplitOptions.RemoveEmptyEntries)
        .Select(s2 => double.Parse(s2))
        .ToArray())
    .ToArray();

var xs = data
    .Select(s => s
        .Take(3)
        .ToArray())
    .ToArray();

var ys = data
    .SelectMany(s => s
        .Skip(3)
        .Take(1))
    .ToList();

LinearRegression r = new LinearRegression();
r.GradientDescentTrain(
    0.000001,
    100000,
    ys,
    xs
    .Select(s => new double[] { 1.0, s[0], s[1], s[2] })
    .ToList());
```

```

var request = File.ReadAllLines("request.csv")
    .Skip(1)
    .Select(s =>
        s.Split(new [] { ';' }, StringSplitOptions.RemoveEmptyEntries)
        .Select(s2 => double.Parse(s2))
        .ToArray())
    .Select(s => new[] { s[0], s[1], s[2], r.Hipotesys(new[] { 1d, s[0], s[1], s[2] }) })
    .OrderBy(o => o[2] * o[3])
    .ToArray();

```

```

double bound = 0;
string solution = "";
Knapsack01.KnapsackBB(
    request
        .Select(s => new Item
        {
            Value = s[2] * s[3],
            Weight = s[2]
        })
        .ToArray(), // items
    request.Count(), // itemIndex
    "", // solution
    300, // maxAmount
    0, // profit
    ref bound, // lowerBound
    ref solution); // bestSolution

```

```

Console.WriteLine("Best solution:");
Console.WriteLine(solution);

```