In [134]:

```
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as sps
import math
import csv
import random
%matplotlib inline
```

In [135]:

```
with open('forestfires.csv') as csvfile:

    # читаем файл
    readCSV = csv.reader(csvfile, delimiter=',')

    # записываем в список
    data = []
    for row in readCSV:
        data.append(row)
print(data[0])
```

```
['X', 'Y', 'month', 'day', 'FFMC', 'DMC', 'DC', 'ISI', 'temp', 'RH',
'wind', 'rain', 'area']
```

Делаем преобразования исходных данных (месяц на индикатор летнего сезона и тд)

In [136]:

```
data = data[1:]

# меняем месяца на индикатор летноге сезона
for i in range(0, len(data), 1):
    if (data[i][2] == 'aug' or data[i][2] == 'jul' or data[i][2] == 'jun'):
        data[i][2] = 1
    else:
        data[i][2] = 0
    # свободный член
    data[i].append(1)

data = np.matrix(data)
data = np.delete(data, 3, 1)

# перемешиваем данные
random.shuffle(data)

# делаем срезку
part1 = round(7*517/10)
regr_model_data = data[:(7*517/10)]

print(data)
```

```
[['7' '5' '0' ..., '0' '0' '1']
 ['7' '5' '0' ..., '0' '0' '1']
 ['7' '4' '0' ..., '0' '0' '1']
 ...,
 ['3' '3' '0' ..., '0' '6.58' '1']
 ['1' '2' '0' ..., '0' '0' '1']
 ['7' '4' '0' ..., '0' '0' '1']]
```

Строим регрессионную модель по первой части

In [137]:

```
# удаляем столбец area
Z = np.delete(regr_model_data, 11, 1)
Z = np.matrix(Z, dtype=float)

# оставляем только area
X = regr_model_data[:,11]
X = np.matrix(X, dtype=float)
```

In [138]:

```
theta_ = np.linalg.inv(Z.T * Z)* Z.T * X
print(theta_)
```

```
[[  7.45512166e-01]
 [  1.58860839e+00]
 [ -9.02729263e+00]
 [  8.11262619e-02]
 [  1.37727165e-01]
 [ -1.34605156e-02]
 [ -4.40156508e-01]
 [  8.42800178e-01]
 [ -4.36893398e-02]
 [  1.01228019e+00]
 [ -1.08819121e+01]
 [ -2.85067651e+01]]
```

Применим модель к остальной части выборки и оценим дисперсию

In [162]:

```
part = round(3*517/10)
data_ = data[part:]

X1 = data_[:,11]
X1 = np.matrix(X1, dtype=float)

Z1 = np.delete(data_, 11, 1)
Z1 = np.matrix(Z1, dtype=float)

sigma2 = 0
for i in range(part):
    sigma2 += (X1[i] - Z1[i]*theta_)**2
sigma2 = math.sqrt(1./part*sigma2)
print(sigma2)
```

```
6.916886637642583
```

Сделаем преобразования для area

In [169]:

```
for c in range(1, 101, 10):
    X_log = np.log(X+c)

    # находим тету
    theta_log = np.linalg.inv(Z.T * Z)* Z.T * X_log

    # применяем к оставшейся
    X_log1 = np.log(X1+c)

    # оценим дисперсию
    sigma2 = 0
    for i in range(part):
        sigma2 += (X_log1[i] - Z1[i]*theta_log)**2
    sigma2 = math.sqrt(1./part*sigma2)

    print(c, "\t", sigma2)
```

```
1       nan
11      0.39237908978775393
21      0.24623093362487442
31      0.18263110090482548
41      0.1459709331282109
51      0.12186608109595903
61      0.10472357917007295
71      0.09187309454591572
81      0.08186629415611506
91      0.07384537417729055
```

```
C:\Program Files\Anaconda3\lib\site-packages\ipykernel\__main__.py:2:
RuntimeWarning: invalid value encountered in log
  from ipykernel import kernelapp as app
```

In [ ]:

```
```