In [307]:
```
import pandas as pd
import numpy as np
import math
import random
```

In [308]:
```
data = pd.read_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/forest-fires/forestfires.csv')
data.head()
```

Out[308]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.0
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.0
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.0
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.0
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.0

Преобразование данных.

Чтобы работать с числовыми координатами нечисловые координаты (month, day) нужно перевести в числовые. Для простоты можно заменить координату month на индикатор летнего сезона, а координату day неиспользовать вообще. По желанию можете сделать преобразование другим способом. Так же желательно добавить координату, тождественно равную единице. Она будет отвечать свободному члену в линейной коомбинации.

In [309]:
```
months = ['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep',
 'oct', 'nov', 'dec']
days = ["mon", "tue", "wed", "thu", "fri", "sat", "sun"]
```

In [310]:
```
data["month"] = data["month"].apply(lambda x: months.index(x))
```

In [311]:
```
data["day"] = data["day"].apply(lambda x: days.index(x))
```

In [312]:
```
data["summer"] = data["month"].apply(lambda x: int(4 < x and x < 8))
```

In [313]:
```
data["ones"] = np.linspace(1,1,data.shape[0])
data.head()
```

Out[313]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	summer	ones
0	7	5	2	4	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.0	0	1.0
1	7	4	9	1	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.0	0	1.0
2	7	4	9	5	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.0	0	1.0
3	8	6	2	4	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.0	0	1.0
4	8	6	2	6	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.0	0	1.0

In [322]:
```
N = data.shape[0]
k = data.shape[1]
```

In [357]:
```
# Для 0.7 выборки
Z = np.array(data)
random.shuffle(Z)
Z = Z[:round(7/10.)*N]
area = np.array(data['area'])[:round(7/10.)*N]
theta=np.dot(np.dot(np.linalg.inv(np.dot(Z.T,Z)), Z.T), area)
```

In [364]:
```
# Для 0.3 выборки
Z = np.array(data)
Z = Z[(round((7/10.)*N)):]
n = len(Z)
area = (np.array(data['area']))[(round((7/10.)*N)):]
sigma2 = (1/(n-k)) * np.dot(area - np.dot(Z,theta), (area-np.dot(Z,theta
)).T);
print('sigma2 = {}'.format(math.sqrt(sigma2)))
```

```
sigma2 = 74.79741368019971
```

In [396]:
```
def trans (c) :
    # Для 0.7 выборки
    Z = np.array(data)
    random.shuffle(Z)
    Z = Z[:round(7/10.)*N]
    area = np.array(data['area'])[:round(7/10.)*N]
    area = np.log(area + c)
    theta=np.dot(np.dot(np.linalg.inv(np.dot(Z.T,Z)), Z.T), area)
    etheta = np.exp(theta) - c

    # Для 0.3 выборки
    Z = np.array(data)
    Z = Z[(round((7/10.)*N)):]
    n = len(Z)
    area = (np.array(data['area']))[(round((7/10.)*N)):]
    area = np.log(area + c)
    sigma2theta = (1/(n-k)) * np.dot(area - np.dot(Z,theta), (area-np.do
t(Z,theta)));
    sigma2etheta = (1/(n-k)) * np.dot(area - np.dot(Z,etheta), (area-np.
dot(Z,etheta)));
    print('sigma2 = {}'.format(math.sqrt(sigma2theta)))
    print('e^sigma2 = {}'.format(math.sqrt(sigma2etheta)))
    print()
```

```
sigma2 = 1.338746151018637
e^sigma2 = 1.5068351396166852
```

In [387]:
```
trans(1)
```

```
sigma2 = 2.519324929595708
e^sigma2 = 1.6989246031857639
```

In [382]:
```
trans(0.5)
```

```
sigma2 = 1.8337968265111493
e^sigma2 = 521.8756058153541
```

In [383]:
```
trans(2)
```

```
sigma2 = 1.3553718259472978
e^sigma2 = 1040.6652634830111
```

In [384]:
```
trans(3)
```

```
sigma2 = 1.581231032062907
e^sigma2 = 2077.1354921559137
```

In [385]:
```
trans(10)
```

```
sigma2 = 1.334996519133072
e^sigma2 = 9377.146462301405
```

In [386]:
```
trans(100)
```

```
sigma2 = 0.2789658823215484
e^sigma2 = 103186.30919530762
```

Видно, что самое подходящее значение c = 1. Разброс при таком c - минимальный.

Также, несложно заметить, что оценка не очень слабо колеблется в зависимости от размешанности выборки:

In [400]:
```
trans(1)
trans(1)
trans(1)
trans(1)
trans(1)
trans(1)
trans(1)
```

```
sigma2 = 1.6261866052876774
e^sigma2 = 1.6289928877924673

sigma2 = 1.4490585804021474
e^sigma2 = 6.090993800983564

sigma2 = 1.8842401257239625
e^sigma2 = 451.3389921165823

sigma2 = 1.617751127520482
e^sigma2 = 2.3750129546630006

sigma2 = 2.6190598902471693
e^sigma2 = 25.510267951116322

sigma2 = 1.5380502839131358
e^sigma2 = 1.811920239810331

sigma2 = 1.5007820179285318
e^sigma2 = 3.6367136040937
```

In [ ]:
```

```