Задача 9.1

In [2]:

```
import numpy as np
import scipy.stats as sps
import pandas as pd
import matplotlib.pyplot as plt
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [3]:

```
frame = pd.read_csv('forestfires.csv', header=0, sep=',')
```

In [4]:

```
frame
```

Out[4]:

	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	0.00
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00
5	8	6	aug	sun	92.3	85.3	488.0	14.7	22.2	29	5.4	0.0	0.00
6	8	6	aug	mon	92.3	88.9	495.6	8.5	24.1	27	3.1	0.0	0.00
7	8	6	aug	mon	91.5	145.4	608.2	10.7	8.0	86	2.2	0.0	0.00
8	8	6	sep	tue	91.0	129.5	692.6	7.0	13.1	63	5.4	0.0	0.00
9	7	5	sep	sat	92.5	88.0	698.6	7.1	22.8	40	4.0	0.0	0.00
10	7	5	sep	sat	92.5	88.0	698.6	7.1	17.8	51	7.2	0.0	0.00
11	7	5	sep	sat	92.8	73.2	713.0	22.6	19.3	38	4.0	0.0	0.00
12	6	5	aug	fri	63.5	70.8	665.3	0.8	17.0	72	6.7	0.0	0.00
13	6	5	sep	mon	90.9	126.5	686.5	7.0	21.3	42	2.2	0.0	0.00
14	6	5	sep	wed	92.9	133.3	699.6	9.2	26.4	21	4.5	0.0	0.00
15	6	5	sep	fri	93.3	141.2	713.9	13.9	22.9	44	5.4	0.0	0.00
16	5	5	mar	sat	91.7	35.8	80.8	7.8	15.1	27	5.4	0.0	0.00
17	8	5	oct	mon	84.9	32.8	664.2	3.0	16.7	47	4.9	0.0	0.00
18	6	4	mar	wed	89.2	27.9	70.8	6.3	15.9	35	4.0	0.0	0.00
19	6	4	apr	sat	86.3	27.4	97.1	5.1	9.3	44	4.5	0.0	0.00
20	6	4	sep	tue	91.0	129.5	692.6	7.0	18.3	40	2.7	0.0	0.00
21	5	4	sep	mon	91.8	78.5	724.3	9.2	19.1	38	2.7	0.0	0.00
22	7	4	jun	sun	94.3	96.3	200.0	56.1	21.0	44	4.5	0.0	0.00
23	7	4	aug	sat	90.2	110.9	537.4	6.2	19.5	43	5.8	0.0	0.00
24	7	4	aug	sat	93.5	139.4	594.2	20.3	23.7	32	5.8	0.0	0.00
25	7	4	aug	sun	91.4	142.4	601.4	10.6	16.3	60	5.4	0.0	0.00
26	7	4	sep	fri	92.4	117.9	668.0	12.2	19.0	34	5.8	0.0	0.00
27	7	4	sep	mon	90.9	126.5	686.5	7.0	19.4	48	1.3	0.0	0.00
28	6	3	sep	sat	93.4	145.4	721.4	8.1	30.2	24	2.7	0.0	0.00

29	6	3	sep	sun	93.5	149.3	728.6	8.1	22.8	39	3.6	0.0	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...
487	5	4	aug	tue	95.1	141.3	605.8	17.7	26.4	34	3.6	0.0	16.40
488	4	4	aug	tue	95.1	141.3	605.8	17.7	19.4	71	7.6	0.0	46.70
489	4	4	aug	wed	95.1	141.3	605.8	17.7	20.6	58	1.3	0.0	0.00
490	4	4	aug	wed	95.1	141.3	605.8	17.7	28.7	33	4.0	0.0	0.00
491	4	4	aug	thu	95.8	152.0	624.1	13.8	32.4	21	4.5	0.0	0.00
492	1	3	aug	fri	95.9	158.0	633.6	11.3	32.4	27	2.2	0.0	0.00
493	1	3	aug	fri	95.9	158.0	633.6	11.3	27.5	29	4.5	0.0	43.32
494	6	6	aug	sat	96.0	164.0	643.0	14.0	30.8	30	4.9	0.0	8.59
495	6	6	aug	mon	96.2	175.5	661.8	16.8	23.9	42	2.2	0.0	0.00
496	4	5	aug	mon	96.2	175.5	661.8	16.8	32.6	26	3.1	0.0	2.77
497	3	4	aug	tue	96.1	181.1	671.2	14.3	32.3	27	2.2	0.0	14.68
498	6	5	aug	tue	96.1	181.1	671.2	14.3	33.3	26	2.7	0.0	40.54
499	7	5	aug	tue	96.1	181.1	671.2	14.3	27.3	63	4.9	6.4	10.82
500	8	6	aug	tue	96.1	181.1	671.2	14.3	21.6	65	4.9	0.8	0.00
501	7	5	aug	tue	96.1	181.1	671.2	14.3	21.6	65	4.9	0.8	0.00
502	4	4	aug	tue	96.1	181.1	671.2	14.3	20.7	69	4.9	0.4	0.00
503	2	4	aug	wed	94.5	139.4	689.1	20.0	29.2	30	4.9	0.0	1.95
504	4	3	aug	wed	94.5	139.4	689.1	20.0	28.9	29	4.9	0.0	49.59
505	1	2	aug	thu	91.0	163.2	744.4	10.1	26.7	35	1.8	0.0	5.80
506	1	2	aug	fri	91.0	166.9	752.6	7.1	18.5	73	8.5	0.0	0.00
507	2	4	aug	fri	91.0	166.9	752.6	7.1	25.9	41	3.6	0.0	0.00
508	1	2	aug	fri	91.0	166.9	752.6	7.1	25.9	41	3.6	0.0	0.00
509	5	4	aug	fri	91.0	166.9	752.6	7.1	21.1	71	7.6	1.4	2.17
510	6	5	aug	fri	91.0	166.9	752.6	7.1	18.2	62	5.4	0.0	0.43
511	8	6	aug	sun	81.6	56.7	665.6	1.9	27.8	35	2.7	0.0	0.00
512	4	3	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44
513	2	4	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29
514	7	4	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16
515	1	4	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00
516	6	3	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00

517 rows × 13 columns

Преобразование данных

In [5]:

```
frame.drop('day', axis=1, inplace=True)
```

In [6]:

```
frame
```

Out[6]:

	X	Y	month	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	mar	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00
1	7	4	oct	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00
2	7	4	oct	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00
3	8	6	mar	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00
4	8	6	mar	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00
5	8	6	aug	92.3	85.3	488.0	14.7	22.2	29	5.4	0.0	0.00
6	8	6	aug	92.3	88.9	495.6	8.5	24.1	27	3.1	0.0	0.00
7	8	6	aug	91.5	145.4	608.2	10.7	8.0	86	2.2	0.0	0.00
8	8	6	sep	91.0	129.5	692.6	7.0	13.1	63	5.4	0.0	0.00
9	7	5	sep	92.5	88.0	698.6	7.1	22.8	40	4.0	0.0	0.00
10	7	5	sep	92.5	88.0	698.6	7.1	17.8	51	7.2	0.0	0.00
11	7	5	sep	92.8	73.2	713.0	22.6	19.3	38	4.0	0.0	0.00
12	6	5	aug	63.5	70.8	665.3	0.8	17.0	72	6.7	0.0	0.00
13	6	5	sep	90.9	126.5	686.5	7.0	21.3	42	2.2	0.0	0.00
14	6	5	sep	92.9	133.3	699.6	9.2	26.4	21	4.5	0.0	0.00
15	6	5	sep	93.3	141.2	713.9	13.9	22.9	44	5.4	0.0	0.00
16	5	5	mar	91.7	35.8	80.8	7.8	15.1	27	5.4	0.0	0.00
17	8	5	oct	84.9	32.8	664.2	3.0	16.7	47	4.9	0.0	0.00
18	6	4	mar	89.2	27.9	70.8	6.3	15.9	35	4.0	0.0	0.00
19	6	4	apr	86.3	27.4	97.1	5.1	9.3	44	4.5	0.0	0.00
20	6	4	sep	91.0	129.5	692.6	7.0	18.3	40	2.7	0.0	0.00
21	5	4	sep	91.8	78.5	724.3	9.2	19.1	38	2.7	0.0	0.00
22	7	4	jun	94.3	96.3	200.0	56.1	21.0	44	4.5	0.0	0.00
23	7	4	aug	90.2	110.9	537.4	6.2	19.5	43	5.8	0.0	0.00
24	7	4	aug	93.5	139.4	594.2	20.3	23.7	32	5.8	0.0	0.00
25	7	4	aug	91.4	142.4	601.4	10.6	16.3	60	5.4	0.0	0.00
26	7	4	sep	92.4	117.9	668.0	12.2	19.0	34	5.8	0.0	0.00
27	7	4	sep	90.9	126.5	686.5	7.0	19.4	48	1.3	0.0	0.00
28	6	3	sep	93.4	145.4	721.4	8.1	30.2	24	2.7	0.0	0.00

29	6	3	sep	93.5	149.3	728.6	8.1	22.8	39	3.6	0.0	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...
487	5	4	aug	95.1	141.3	605.8	17.7	26.4	34	3.6	0.0	16.40
488	4	4	aug	95.1	141.3	605.8	17.7	19.4	71	7.6	0.0	46.70
489	4	4	aug	95.1	141.3	605.8	17.7	20.6	58	1.3	0.0	0.00
490	4	4	aug	95.1	141.3	605.8	17.7	28.7	33	4.0	0.0	0.00
491	4	4	aug	95.8	152.0	624.1	13.8	32.4	21	4.5	0.0	0.00
492	1	3	aug	95.9	158.0	633.6	11.3	32.4	27	2.2	0.0	0.00
493	1	3	aug	95.9	158.0	633.6	11.3	27.5	29	4.5	0.0	43.32
494	6	6	aug	96.0	164.0	643.0	14.0	30.8	30	4.9	0.0	8.59
495	6	6	aug	96.2	175.5	661.8	16.8	23.9	42	2.2	0.0	0.00
496	4	5	aug	96.2	175.5	661.8	16.8	32.6	26	3.1	0.0	2.77
497	3	4	aug	96.1	181.1	671.2	14.3	32.3	27	2.2	0.0	14.68
498	6	5	aug	96.1	181.1	671.2	14.3	33.3	26	2.7	0.0	40.54
499	7	5	aug	96.1	181.1	671.2	14.3	27.3	63	4.9	6.4	10.82
500	8	6	aug	96.1	181.1	671.2	14.3	21.6	65	4.9	0.8	0.00
501	7	5	aug	96.1	181.1	671.2	14.3	21.6	65	4.9	0.8	0.00
502	4	4	aug	96.1	181.1	671.2	14.3	20.7	69	4.9	0.4	0.00
503	2	4	aug	94.5	139.4	689.1	20.0	29.2	30	4.9	0.0	1.95
504	4	3	aug	94.5	139.4	689.1	20.0	28.9	29	4.9	0.0	49.59
505	1	2	aug	91.0	163.2	744.4	10.1	26.7	35	1.8	0.0	5.80
506	1	2	aug	91.0	166.9	752.6	7.1	18.5	73	8.5	0.0	0.00
507	2	4	aug	91.0	166.9	752.6	7.1	25.9	41	3.6	0.0	0.00
508	1	2	aug	91.0	166.9	752.6	7.1	25.9	41	3.6	0.0	0.00
509	5	4	aug	91.0	166.9	752.6	7.1	21.1	71	7.6	1.4	2.17
510	6	5	aug	91.0	166.9	752.6	7.1	18.2	62	5.4	0.0	0.43
511	8	6	aug	81.6	56.7	665.6	1.9	27.8	35	2.7	0.0	0.00
512	4	3	aug	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44
513	2	4	aug	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29
514	7	4	aug	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16
515	1	4	aug	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00
516	6	3	nov	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00

517 rows × 12 columns

In [7]:

```
months = np.array(['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oc
for i in range(months.size):
    frame.loc[frame['month'] == months[i],'month'] = i + 1
```

In [8]:

```
frame
```

Out[8]:

	X	Y	month	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	3	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00
1	7	4	10	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00
2	7	4	10	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00
3	8	6	3	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00
4	8	6	3	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00
5	8	6	8	92.3	85.3	488.0	14.7	22.2	29	5.4	0.0	0.00
6	8	6	8	92.3	88.9	495.6	8.5	24.1	27	3.1	0.0	0.00
7	8	6	8	91.5	145.4	608.2	10.7	8.0	86	2.2	0.0	0.00
8	8	6	9	91.0	129.5	692.6	7.0	13.1	63	5.4	0.0	0.00
9	7	5	9	92.5	88.0	698.6	7.1	22.8	40	4.0	0.0	0.00
10	7	5	9	92.5	88.0	698.6	7.1	17.8	51	7.2	0.0	0.00
11	7	5	9	92.8	73.2	713.0	22.6	19.3	38	4.0	0.0	0.00
12	6	5	8	63.5	70.8	665.3	0.8	17.0	72	6.7	0.0	0.00
13	6	5	9	90.9	126.5	686.5	7.0	21.3	42	2.2	0.0	0.00
14	6	5	9	92.9	133.3	699.6	9.2	26.4	21	4.5	0.0	0.00
15	6	5	9	93.3	141.2	713.9	13.9	22.9	44	5.4	0.0	0.00
16	5	5	3	91.7	35.8	80.8	7.8	15.1	27	5.4	0.0	0.00
17	8	5	10	84.9	32.8	664.2	3.0	16.7	47	4.9	0.0	0.00
18	6	4	3	89.2	27.9	70.8	6.3	15.9	35	4.0	0.0	0.00
19	6	4	4	86.3	27.4	97.1	5.1	9.3	44	4.5	0.0	0.00
20	6	4	9	91.0	129.5	692.6	7.0	18.3	40	2.7	0.0	0.00
21	5	4	9	91.8	78.5	724.3	9.2	19.1	38	2.7	0.0	0.00
22	7	4	6	94.3	96.3	200.0	56.1	21.0	44	4.5	0.0	0.00
23	7	4	8	90.2	110.9	537.4	6.2	19.5	43	5.8	0.0	0.00
24	7	4	8	93.5	139.4	594.2	20.3	23.7	32	5.8	0.0	0.00
25	7	4	8	91.4	142.4	601.4	10.6	16.3	60	5.4	0.0	0.00
26	7	4	9	92.4	117.9	668.0	12.2	19.0	34	5.8	0.0	0.00
27	7	4	9	90.9	126.5	686.5	7.0	19.4	48	1.3	0.0	0.00
28	6	3	9	93.4	145.4	721.4	8.1	30.2	24	2.7	0.0	0.00

29	6	3	9	93.5	149.3	728.6	8.1	22.8	39	3.6	0.0	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...
487	5	4	8	95.1	141.3	605.8	17.7	26.4	34	3.6	0.0	16.40
488	4	4	8	95.1	141.3	605.8	17.7	19.4	71	7.6	0.0	46.70
489	4	4	8	95.1	141.3	605.8	17.7	20.6	58	1.3	0.0	0.00
490	4	4	8	95.1	141.3	605.8	17.7	28.7	33	4.0	0.0	0.00
491	4	4	8	95.8	152.0	624.1	13.8	32.4	21	4.5	0.0	0.00
492	1	3	8	95.9	158.0	633.6	11.3	32.4	27	2.2	0.0	0.00
493	1	3	8	95.9	158.0	633.6	11.3	27.5	29	4.5	0.0	43.32
494	6	6	8	96.0	164.0	643.0	14.0	30.8	30	4.9	0.0	8.59
495	6	6	8	96.2	175.5	661.8	16.8	23.9	42	2.2	0.0	0.00
496	4	5	8	96.2	175.5	661.8	16.8	32.6	26	3.1	0.0	2.77
497	3	4	8	96.1	181.1	671.2	14.3	32.3	27	2.2	0.0	14.68
498	6	5	8	96.1	181.1	671.2	14.3	33.3	26	2.7	0.0	40.54
499	7	5	8	96.1	181.1	671.2	14.3	27.3	63	4.9	6.4	10.82
500	8	6	8	96.1	181.1	671.2	14.3	21.6	65	4.9	0.8	0.00
501	7	5	8	96.1	181.1	671.2	14.3	21.6	65	4.9	0.8	0.00
502	4	4	8	96.1	181.1	671.2	14.3	20.7	69	4.9	0.4	0.00
503	2	4	8	94.5	139.4	689.1	20.0	29.2	30	4.9	0.0	1.95
504	4	3	8	94.5	139.4	689.1	20.0	28.9	29	4.9	0.0	49.59
505	1	2	8	91.0	163.2	744.4	10.1	26.7	35	1.8	0.0	5.80
506	1	2	8	91.0	166.9	752.6	7.1	18.5	73	8.5	0.0	0.00
507	2	4	8	91.0	166.9	752.6	7.1	25.9	41	3.6	0.0	0.00
508	1	2	8	91.0	166.9	752.6	7.1	25.9	41	3.6	0.0	0.00
509	5	4	8	91.0	166.9	752.6	7.1	21.1	71	7.6	1.4	2.17
510	6	5	8	91.0	166.9	752.6	7.1	18.2	62	5.4	0.0	0.43
511	8	6	8	81.6	56.7	665.6	1.9	27.8	35	2.7	0.0	0.00
512	4	3	8	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44
513	2	4	8	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29
514	7	4	8	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16
515	1	4	8	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00
516	6	3	11	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00

517 rows × 12 columns

In [9]:

```
frame.dtypes
```

Out[9]:

```
X          int64
Y          int64
month     object
FFMC     float64
DMC      float64
DC       float64
ISI      float64
temp     float64
RH         int64
wind     float64
rain     float64
area     float64
dtype: object
```

In [10]:

```
frame['month'] = frame['month'].astype(int)
```

In [11]:

```
frame.dtypes
```

Out[11]:

```
X          int64
Y          int64
month      int32
FFMC     float64
DMC      float64
DC       float64
ISI      float64
temp     float64
RH         int64
wind     float64
rain     float64
area     float64
dtype: object
```

In [12]:

```
frame['constant term'] = 1
```

In [13]:

```
frame
```

Out[13]:

	X	Y	month	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	constar term
0	7	5	3	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00	1
1	7	4	10	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00	1
2	7	4	10	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00	1
3	8	6	3	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00	1
4	8	6	3	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00	1
5	8	6	8	92.3	85.3	488.0	14.7	22.2	29	5.4	0.0	0.00	1
6	8	6	8	92.3	88.9	495.6	8.5	24.1	27	3.1	0.0	0.00	1
7	8	6	8	91.5	145.4	608.2	10.7	8.0	86	2.2	0.0	0.00	1
8	8	6	9	91.0	129.5	692.6	7.0	13.1	63	5.4	0.0	0.00	1
9	7	5	9	92.5	88.0	698.6	7.1	22.8	40	4.0	0.0	0.00	1
10	7	5	9	92.5	88.0	698.6	7.1	17.8	51	7.2	0.0	0.00	1
11	7	5	9	92.8	73.2	713.0	22.6	19.3	38	4.0	0.0	0.00	1
12	6	5	8	63.5	70.8	665.3	0.8	17.0	72	6.7	0.0	0.00	1
13	6	5	9	90.9	126.5	686.5	7.0	21.3	42	2.2	0.0	0.00	1
14	6	5	9	92.9	133.3	699.6	9.2	26.4	21	4.5	0.0	0.00	1
15	6	5	9	93.3	141.2	713.9	13.9	22.9	44	5.4	0.0	0.00	1
16	5	5	3	91.7	35.8	80.8	7.8	15.1	27	5.4	0.0	0.00	1
17	8	5	10	84.9	32.8	664.2	3.0	16.7	47	4.9	0.0	0.00	1
18	6	4	3	89.2	27.9	70.8	6.3	15.9	35	4.0	0.0	0.00	1
19	6	4	4	86.3	27.4	97.1	5.1	9.3	44	4.5	0.0	0.00	1
20	6	4	9	91.0	129.5	692.6	7.0	18.3	40	2.7	0.0	0.00	1
21	5	4	9	91.8	78.5	724.3	9.2	19.1	38	2.7	0.0	0.00	1
22	7	4	6	94.3	96.3	200.0	56.1	21.0	44	4.5	0.0	0.00	1
23	7	4	8	90.2	110.9	537.4	6.2	19.5	43	5.8	0.0	0.00	1
24	7	4	8	93.5	139.4	594.2	20.3	23.7	32	5.8	0.0	0.00	1
25	7	4	8	91.4	142.4	601.4	10.6	16.3	60	5.4	0.0	0.00	1
26	7	4	9	92.4	117.9	668.0	12.2	19.0	34	5.8	0.0	0.00	1
27	7	4	9	90.9	126.5	686.5	7.0	19.4	48	1.3	0.0	0.00	1

28	6	3	9	93.4	145.4	721.4	8.1	30.2	24	2.7	0.0	0.00	1
29	6	3	9	93.5	149.3	728.6	8.1	22.8	39	3.6	0.0	0.00	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
487	5	4	8	95.1	141.3	605.8	17.7	26.4	34	3.6	0.0	16.40	1
488	4	4	8	95.1	141.3	605.8	17.7	19.4	71	7.6	0.0	46.70	1
489	4	4	8	95.1	141.3	605.8	17.7	20.6	58	1.3	0.0	0.00	1
490	4	4	8	95.1	141.3	605.8	17.7	28.7	33	4.0	0.0	0.00	1
491	4	4	8	95.8	152.0	624.1	13.8	32.4	21	4.5	0.0	0.00	1
492	1	3	8	95.9	158.0	633.6	11.3	32.4	27	2.2	0.0	0.00	1
493	1	3	8	95.9	158.0	633.6	11.3	27.5	29	4.5	0.0	43.32	1
494	6	6	8	96.0	164.0	643.0	14.0	30.8	30	4.9	0.0	8.59	1
495	6	6	8	96.2	175.5	661.8	16.8	23.9	42	2.2	0.0	0.00	1
496	4	5	8	96.2	175.5	661.8	16.8	32.6	26	3.1	0.0	2.77	1
497	3	4	8	96.1	181.1	671.2	14.3	32.3	27	2.2	0.0	14.68	1
498	6	5	8	96.1	181.1	671.2	14.3	33.3	26	2.7	0.0	40.54	1
499	7	5	8	96.1	181.1	671.2	14.3	27.3	63	4.9	6.4	10.82	1
500	8	6	8	96.1	181.1	671.2	14.3	21.6	65	4.9	0.8	0.00	1
501	7	5	8	96.1	181.1	671.2	14.3	21.6	65	4.9	0.8	0.00	1
502	4	4	8	96.1	181.1	671.2	14.3	20.7	69	4.9	0.4	0.00	1
503	2	4	8	94.5	139.4	689.1	20.0	29.2	30	4.9	0.0	1.95	1
504	4	3	8	94.5	139.4	689.1	20.0	28.9	29	4.9	0.0	49.59	1
505	1	2	8	91.0	163.2	744.4	10.1	26.7	35	1.8	0.0	5.80	1
506	1	2	8	91.0	166.9	752.6	7.1	18.5	73	8.5	0.0	0.00	1
507	2	4	8	91.0	166.9	752.6	7.1	25.9	41	3.6	0.0	0.00	1
508	1	2	8	91.0	166.9	752.6	7.1	25.9	41	3.6	0.0	0.00	1
509	5	4	8	91.0	166.9	752.6	7.1	21.1	71	7.6	1.4	2.17	1
510	6	5	8	91.0	166.9	752.6	7.1	18.2	62	5.4	0.0	0.43	1
511	8	6	8	81.6	56.7	665.6	1.9	27.8	35	2.7	0.0	0.00	1
512	4	3	8	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44	1
513	2	4	8	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29	1
514	7	4	8	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16	1
515	1	4	8	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00	1
516	6	3	11	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00	1

517 rows × 13 columns

In [14]:

```
data = pd.DataFrame.as_matrix(frame)
np.random.shuffle(data)
```

Разобьём выборку на 2 части в отношении 7:3 :

In [15]:

```
517 * 0.7
```

Out[15]:

361.9

In [16]:

```
first_part = data[:362]
second_part = data[362:]
```

Построим регрессионную модель по первой части выборки. Линейная регрессионная модель - $X = Z\theta + \varepsilon$. В этом случае X - столбец area, Z - матрица, столбцы которой - другие данные, θ - коэффициенты линейной комбинации.

In [17]:

```
def make_pair(x,y):
    return lambda n: x if n==0 else y

def first(p):
    return p(0)

def second(p):
    return p(1)

def find_params(part):
    x = part[:, part[0].size - 2].reshape(part[:, 0].size, 1)
    z = np.empty((part[:, 0].size, part[0].size - 1))
    z[:, :z[0].size - 1] = part[:, :z[0].size - 1]
    z[:,z[0].size - 1] = part[:, part[0].size - 1]
    return make_pair(x, z)
```

In [85]:

```
p = find_params(first_part)
x = first(p)
z = second(p)
```

Тогда оценка наименьших квадратов параметра θ - $\widehat{\theta} = (Z^T Z)^{-1} Z^T X$. Найдем эту оценку:

In [86]:

```
def find_theta_est(x, z):
    return np.linalg.inv(z.T.dot(z)).dot(z.T).dot(x)
```

In [87]:

```
theta_est = find_theta_est(x, z)
```

Применим модель ко второй части выборки:

In [21]:

```
p = find_params(second_part)
real_x = first(p)
z2 = second(p)
x_est = z2.dot(theta_est)
```

Среднеквадратичная ошибка - $\sigma = \sqrt{\frac{\sum_{i=1}^{n} \left(X_i - \widehat{X}\right)^2}{n}}$, где X_i - реальное значение, $\widehat{X}$ - предсказанное.

In [22]:

```
from math import sqrt
```

In [41]:

```
def find_standart_deviation(x_est, real_x):
    return sqrt(((x_est - real_x)**2).sum() / x_est.size)
```

Среднеквадратичная ошибка для второй части выборки:

In [43]:

```
find_standart_deviation(x_est, real_x)
```

Out[43]:

88.73072446791205

Регрессионная модель для $ln(c + x)$

In [44]:

```
from math import log
```

Среднеквадратичная ошибка для преобразованных значений

Построим линейную регрессионную модель, найдем оценку наименьших квадратов параметра θ и среднеквадратичную ошибку для преобразованных значений:

In [45]:

```
n = 100
```

In [80]:

```
p = find_params(first_part)
x = first(p)
z = second(p)
c = np.arange(1, n + 1, 1)
theta_est1 = np.empty((12, n))
for i in range(n):
    x1 = np.log(x + c[i])
    theta_est1[:, i] = np.linalg.inv(z.T.dot(z)).dot(z.T).dot(x1).reshape(12)
```

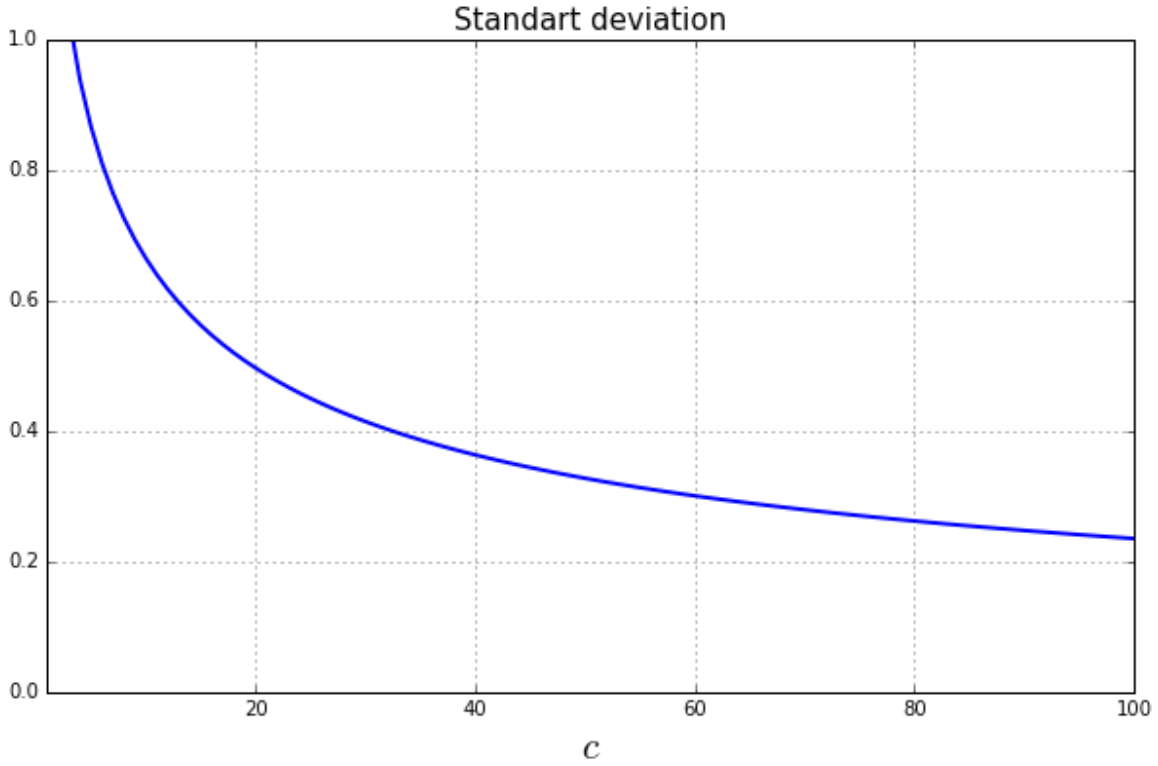
In [81]:

```
p2 = find_params(second_part)
x2 = first(p2)
z2 = second(p2)
x_est = np.empty((x2.size, n))
st_dev2 = np.empty(n)
for i in range(n):
    x_est[:, i] = z2.dot(theta_est1[:, i])
    st_dev2[i] = np.std(x_est[:, i] - np.log(x2 + c[i]))
```

In [82]:

```
plt.figure(figsize=(10, 6))

plt.plot(c, st_dev2, linewidth = 2)
plt.xlim(1, n)
plt.ylim(0, 1)
plt.xlabel('$c$', fontsize = 20)
plt.title('Standart deviation', fontsize = 15)
plt.grid()
plt.show()
```



Среднеквадратичная ошибка для исходных значений

Для нахождения среднеквадратичной ошибки для исходных данных, применим к оценкам обратное преобразование - $f^{-1}(y) = e^y - c$:

In [77]:

```
res = np.empty(n)
for i in range(n):
    x_est[:, i] = np.exp(x_est[:, i]) - c[i]
    res[i] = np.std(x_est[:, i] - x2)
```

In [78]:

```
plt.figure(figsize=(10, 6))

plt.plot(c, res, linewidth = 2)
plt.xlim(1, n)
#plt.ylim(0, 1000)
plt.xlabel('$c$', fontsize = 20)
plt.title('Standart deviation', fontsize = 15)
plt.grid()
plt.show()
```