Яндекс

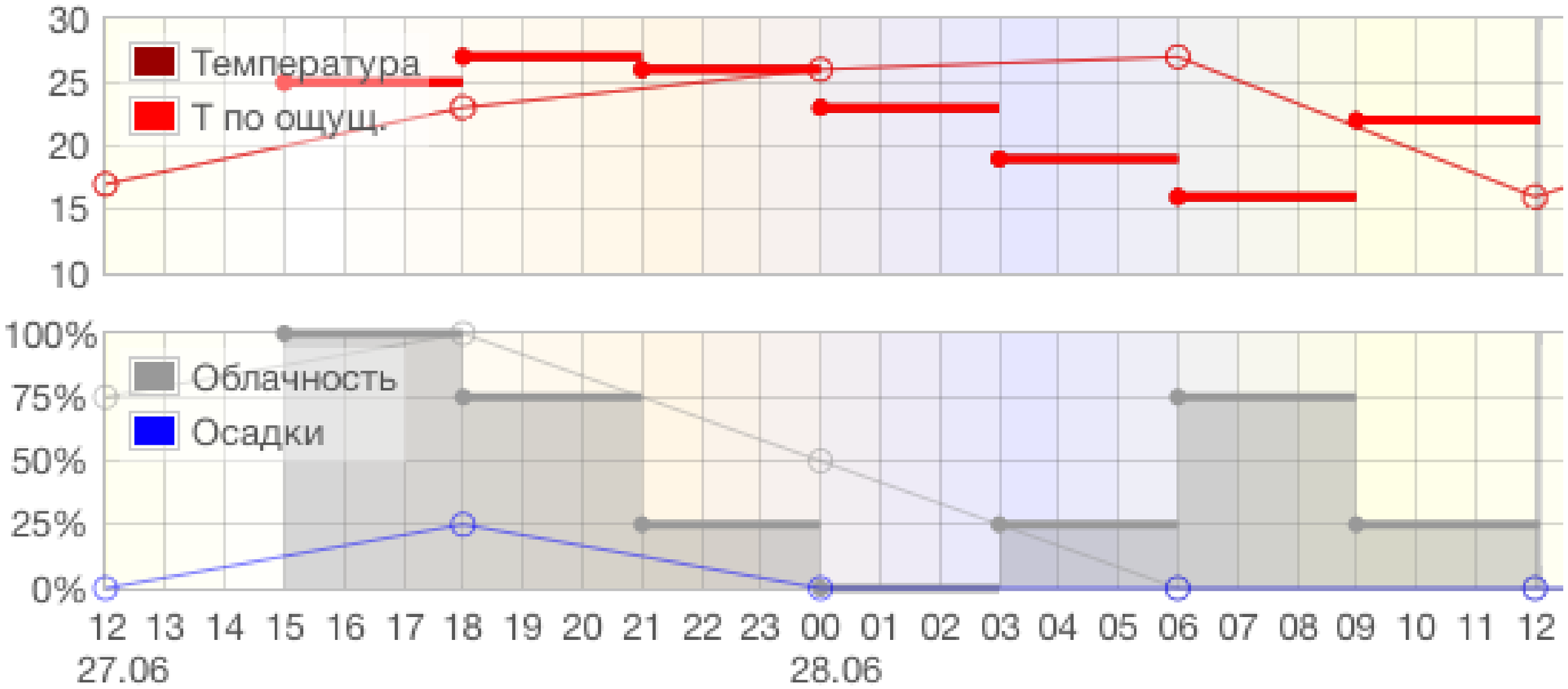
Python и вычисления (в Яндекс.Погоде)

Руслан Гроховецкий
руководитель группы справочных сервисов
Я.Субботник в Екатеринбурге, 06.07.13

Что вычисляем

Анализ данных
в Яндекс.Погоде

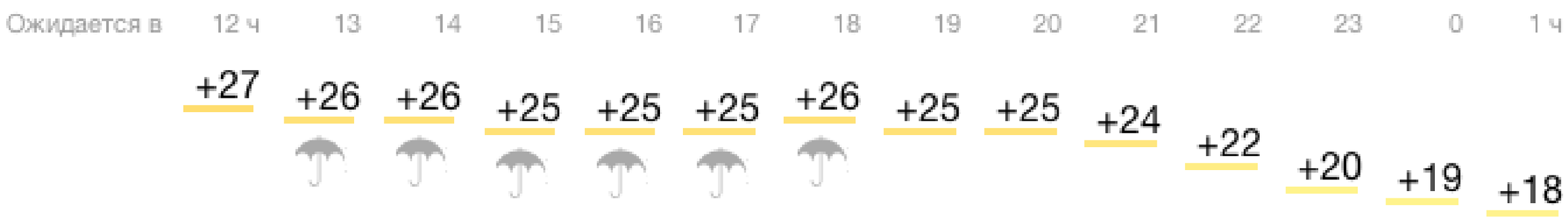
Яндекс не занимается гидрометеорологической деятельностью



Станция делает наблюдения раз в 3 часа

В Екатеринбурге +25 °C

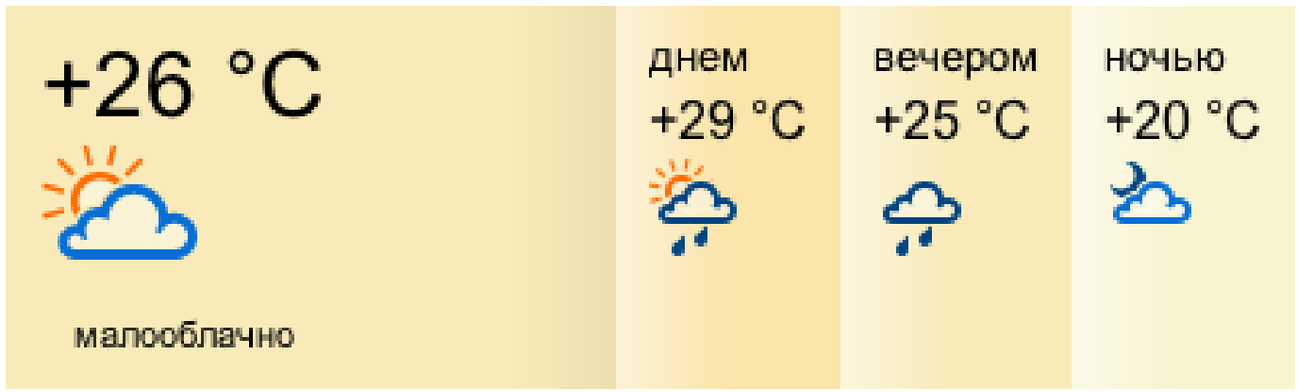
Ветер 5 м/с, ясно



Прогноз есть на каждый час

Череповец другой город Погода на карте

сейчас



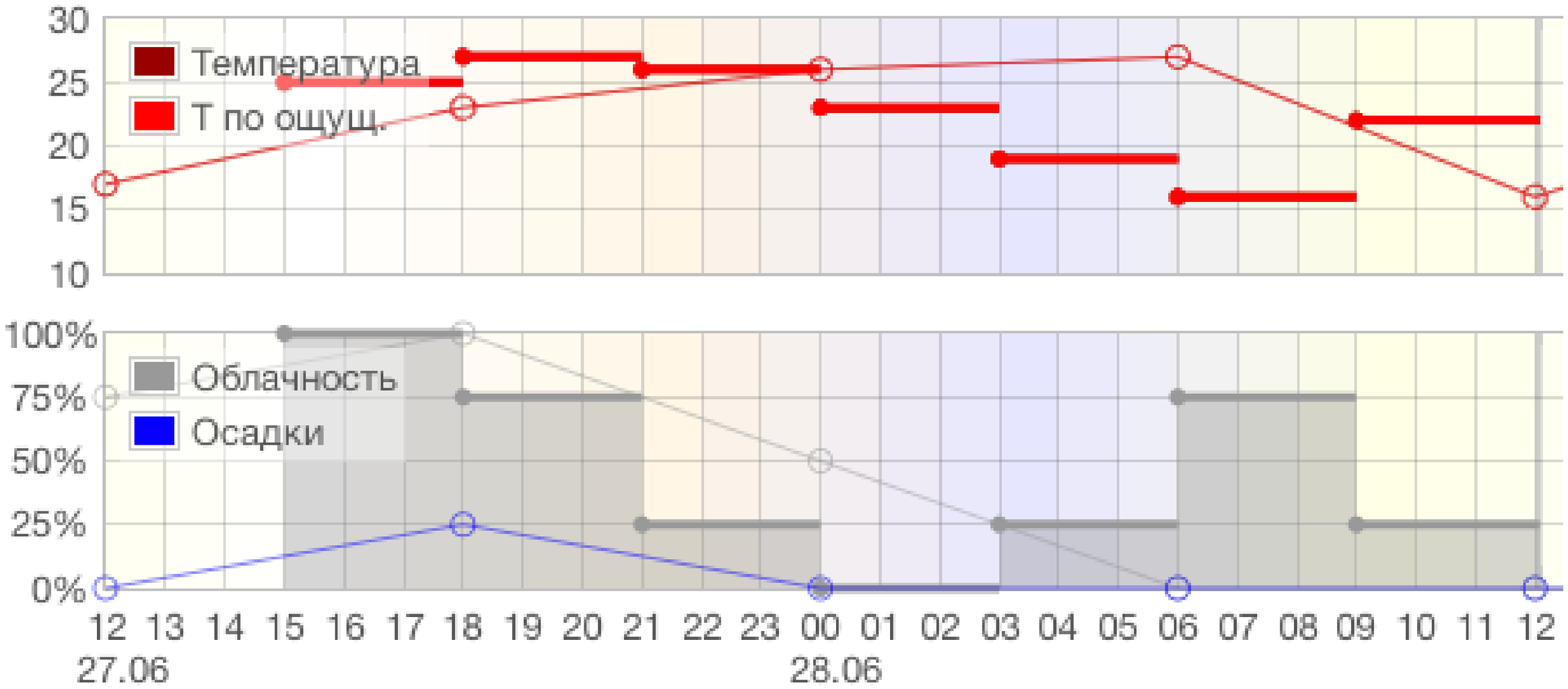
Давление: 751 мм рт. ст.
Ветер: восточный, 0.4 м/с (1.4 км/ч)
Влажность: 71%
Восход: 04:16 Заход: 22:46

Данные зарегистрированы недавно

На основе прогноза

Погодные данные на основе прогноза

Когда вместо устаревшего наблюдения показывать прогноз?



Станция делает наблюдения раз в 3 часа

Как сильно меняется температура со временем?

Как вычисляем

NumPy + SciPy + matplotlib
ipython --pylab

Python — не числодробилка

- Скриптовая природа;
- простой и элегантный синтаксис;
- высокоуровневые структуры данных;
- интерактивность IPython;
- нет числодробительной силы (.

Нужно добавить числодробительной силы!

А где ее взять?

NumPy

- Это расширение CPython
- Поддержка больших массивов и матриц
- Операции над массивами, индексация
- Математические функции, алгоритмы
- Заменяет собой MATLAB
- Много дополняющих пакетов
  - SciPy+SciKits, Matplotlib, PyTables, Pandas, StatsModels, Sympy...
- Использует LAPACK (числодроб. сила!)
  - Который использует BLAS/ATLAS, написанный на C и Fortran

Pylab

- Matplotlib — 2D графики
- NumPy
- IPython

```
$ ipython --pylab

Welcome to pylab, a matplotlib-based Python environment
For more information, type 'help(pylab)'.
```

Данные

- 12173 города
- ≥20000 станций (метеостанции и аэропорты)
- ≈200 млн. накопленных наблюдений

Наблюдения за температурой

Есть данные наблюдений на метеостанциях, выгруженные из архивной БД.

```
# timestamp; Temperature
1360270800; 18
1360269000; 18
1360267200; 18
1360265400; 18
1360263600; 18
1360261800; 19
1360260000; 19
1360258200; 19
1360256400; 19
1360254600; 19
1360252800; 19
1360251000; 19
1360249200; 20
1360247400; 20
1360245600; 20
1360243800; 20
...
```

Берем в руки pylab

```
>>> data = loadtxt('sheremetjevo.csv', delimiter=';', dtype=int32)
>>> data
array([[1334043000,          2],
       [1334044800,          2],
       [1334046600,          3],
       ...,
       [1372793400,         19],
       [1372795200,         18],
       [1372797000,         17]], dtype=int32)
```

Берем в руки pylab

```
>>> data = loadtxt('sheremetjevo.csv', delimiter=';', dtype=int32)
>>> data
array([[1334043000,          2],
       [1334044800,          2],
       [1334046600,          3],
       ...,
       [1372793400,         19],
       [1372795200,         18],
       [1372797000,         17]], dtype=int32)

>>> data.shape
(21022, 2)
```

Берем в руки pylab

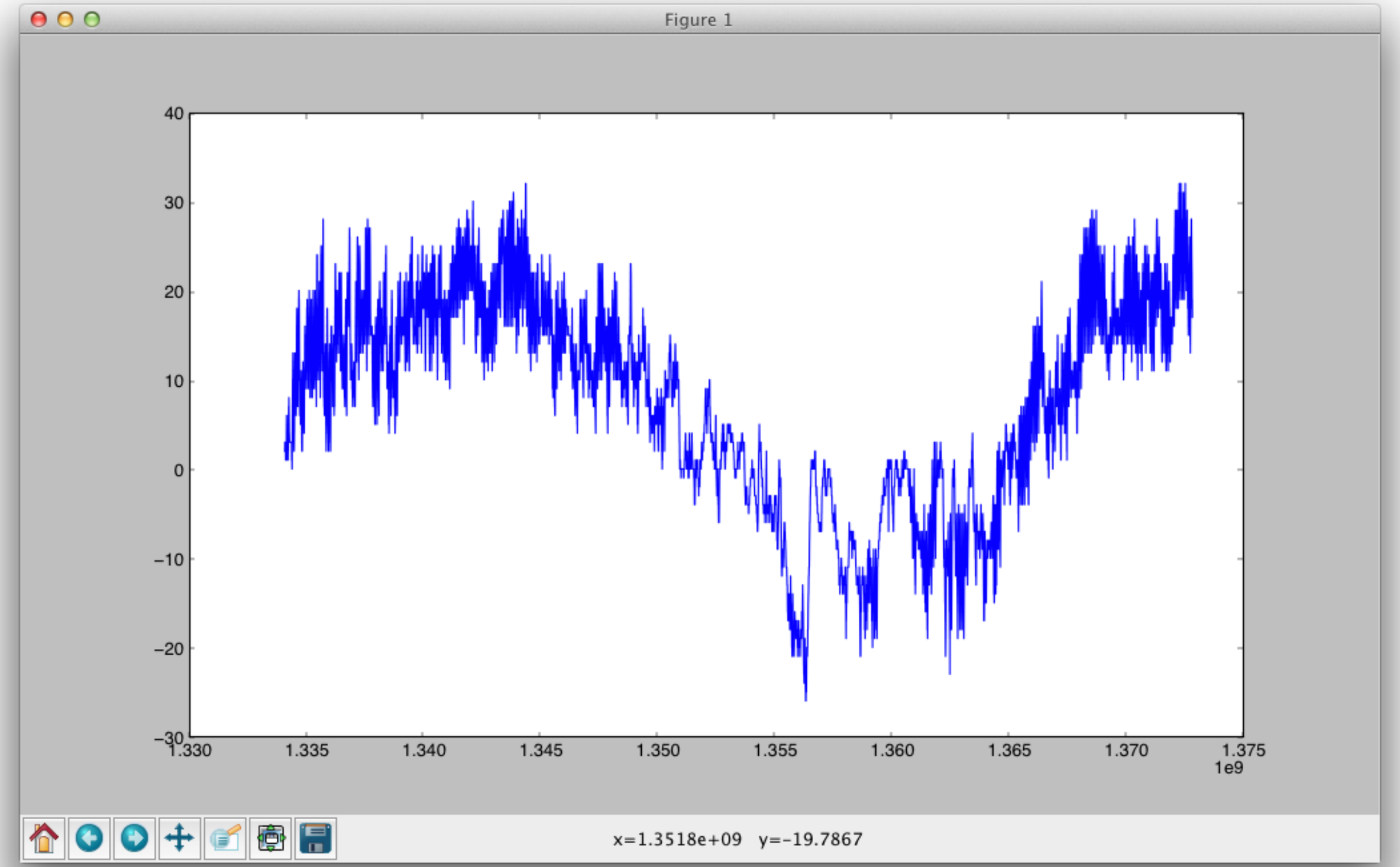
```
>>> data = loadtxt('sheremetjevo.csv', delimiter=';', dtype=int32)
>>> data
array([[1334043000,          2],
       [1334044800,          2],
       [1334046600,          3],
       ...,
       [1372793400,         19],
       [1372795200,         18],
       [1372797000,         17]], dtype=int32)

>>> data.shape
(21022, 2)

>>> times, temps = data[:, 0], data[:, 1]
```

Берем в руки pylab

```
>>> data = loadtxt('sheremetjevo.csv', delimiter=';', dtype=int32)
>>> data
array([[1334043000,           2],
       [1334044800,           2],
       [1334046600,           3],
       ...,
       [1372793400,          19],
       [1372795200,          18],
       [1372797000,          17]], dtype=int32)

>>> data.shape
(21022, 2)

>>> times, temps = data[:, 0], data[:, 1]
>>> temps.max(), temps.min()
(32, -26)
```

Берем в руки pylab

```
>>> data = loadtxt('sheremetjevo.csv', delimiter=';', dtype=int32)
>>> data
array([[1334043000,           2],
       [1334044800,           2],
       [1334046600,           3],
       ...,
       [1372793400,          19],
       [1372795200,          18],
       [1372797000,          17]], dtype=int32)

>>> data.shape
(21022, 2)

>>> times, temps = data[:, 0], data[:, 1]
>>> temps.max(), temps.min()
(32, -26)

>>> plot(times, temps)
```



>>> plot(times, temps)

Векторизация

- Одна операция — много данных
- Нет циклов на Python-е

Разность температур соседних наблюдений

```
# python without numpy

>>> [temps[i+1] - temps[i]
     for i in xrange(len(temps)-1)]
```

```
# numpy

>>> temps[1:] - temps[:-1]
```

Векторизация

- Одна операция — много данных
- Нет циклов на Python-е

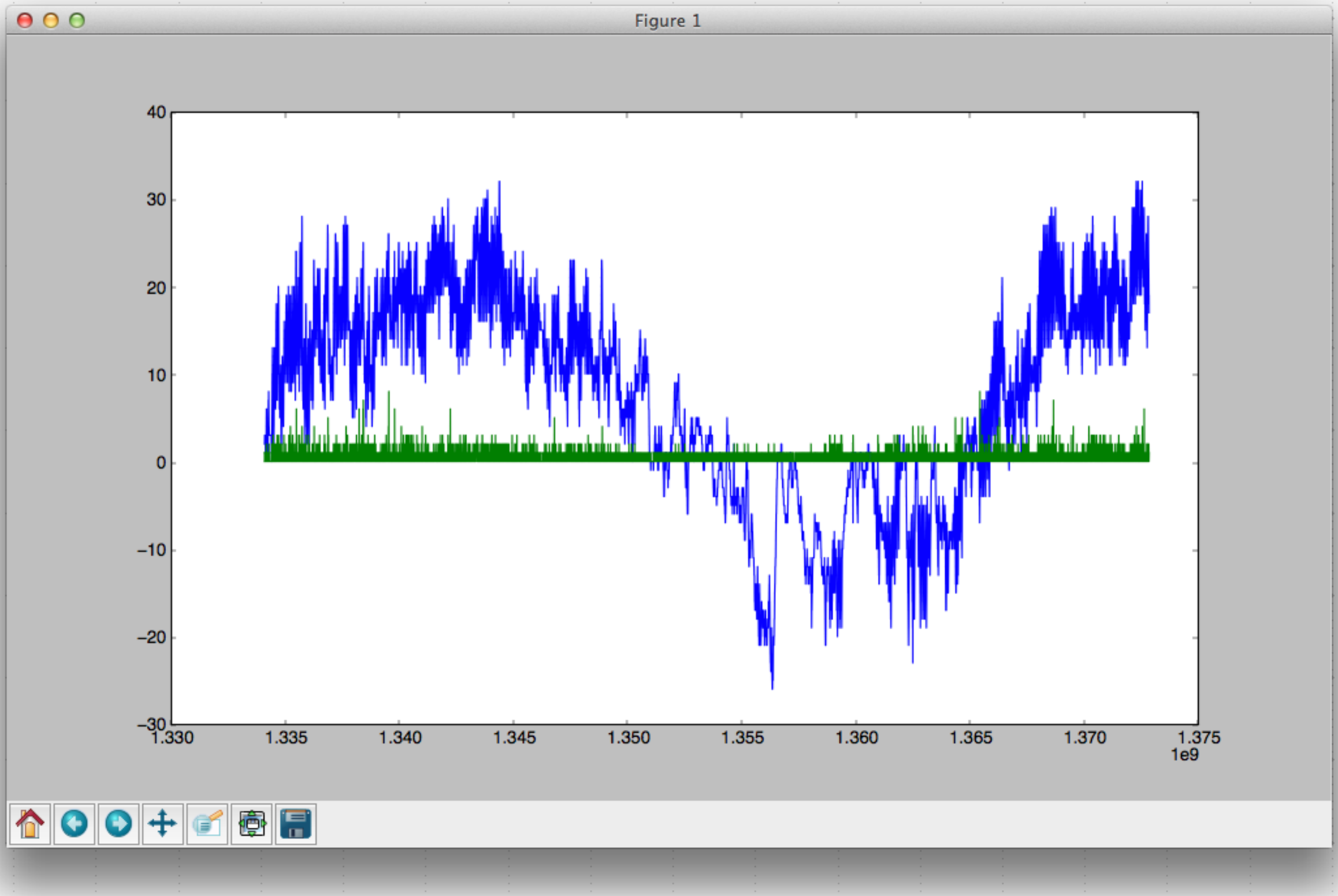
Разность температур соседних наблюдений

```
# python without numpy

>>> [temps[i+1] - temps[i]
     for i in xrange(len(temps)-1)]
```

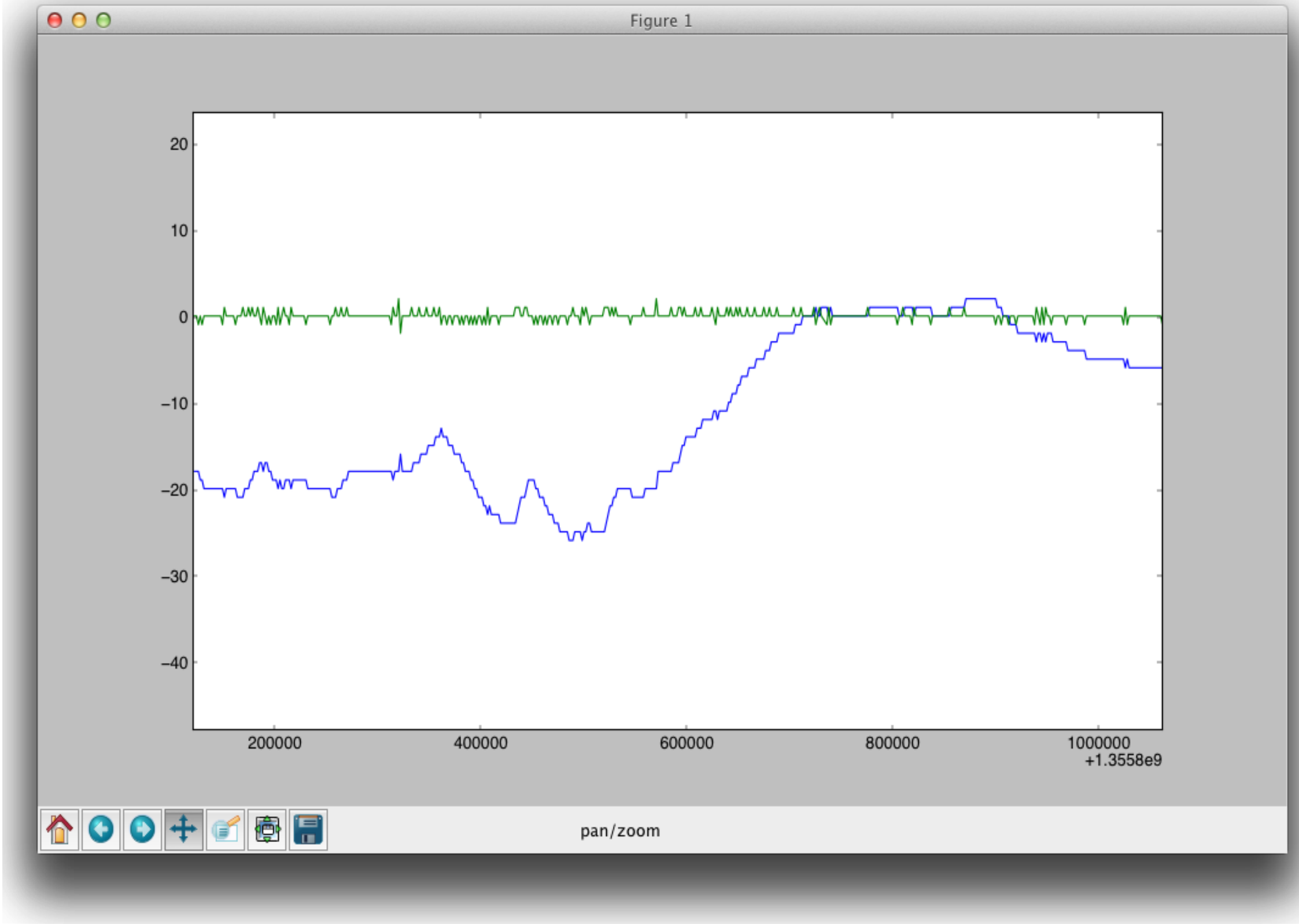
```
# numpy

>>> temps[1:] - temps[:-1]
>>> diff(temps) # то же самое
```

Разности температур между замерами

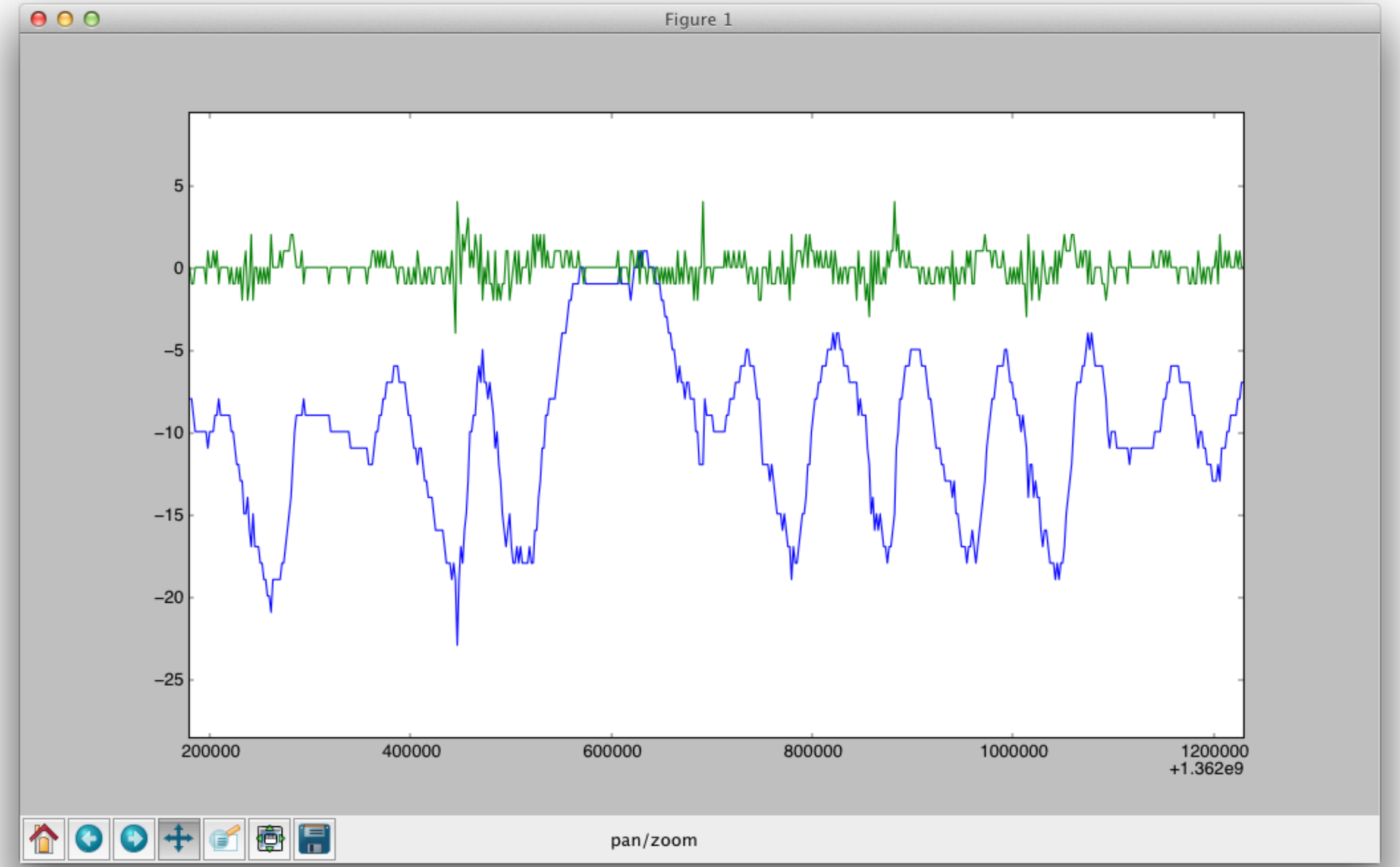
```
>>> temp_diff = absolute(diff(temps))

>>> temp_diff.max()
8.0

>>> plot(times[1:], temp_diff)
```




```
>>> plot(times[1:], temp_diff)
```



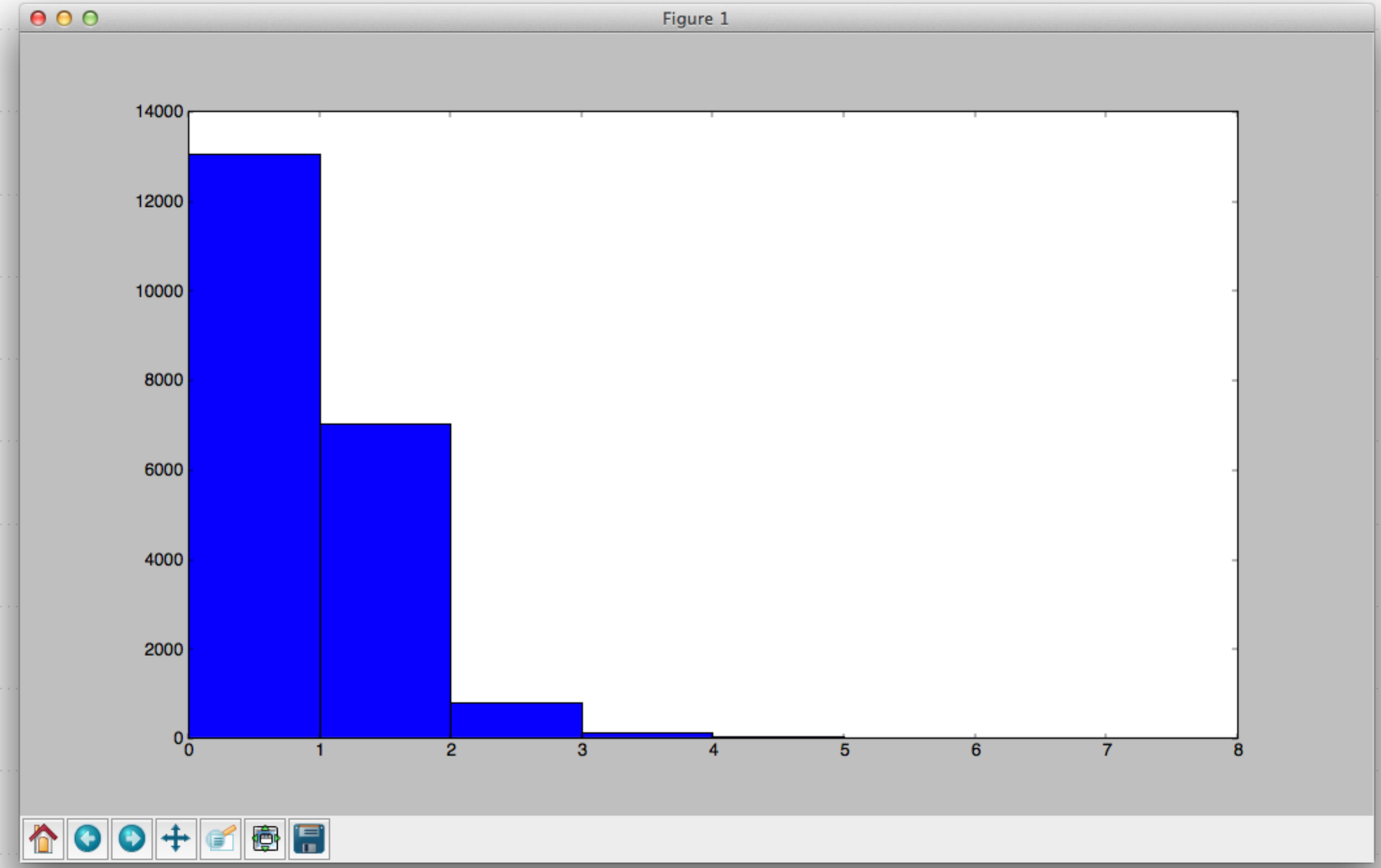
plot(times[:-1], temp_diff)



plot(times[:-1], temp_diff)

Строим гистограмму

```
>>> histogram(temp_diff, bins=range(0,9))

(array([13044, 7028,  795,  118,  21,   5,   6,   4]),
 Array([   0,    1,    2,    3,   4,   5,   6,   7,   8]))

>>> hist, scale = _

>>> hist(temp_diff, bins=range(0,9))
```



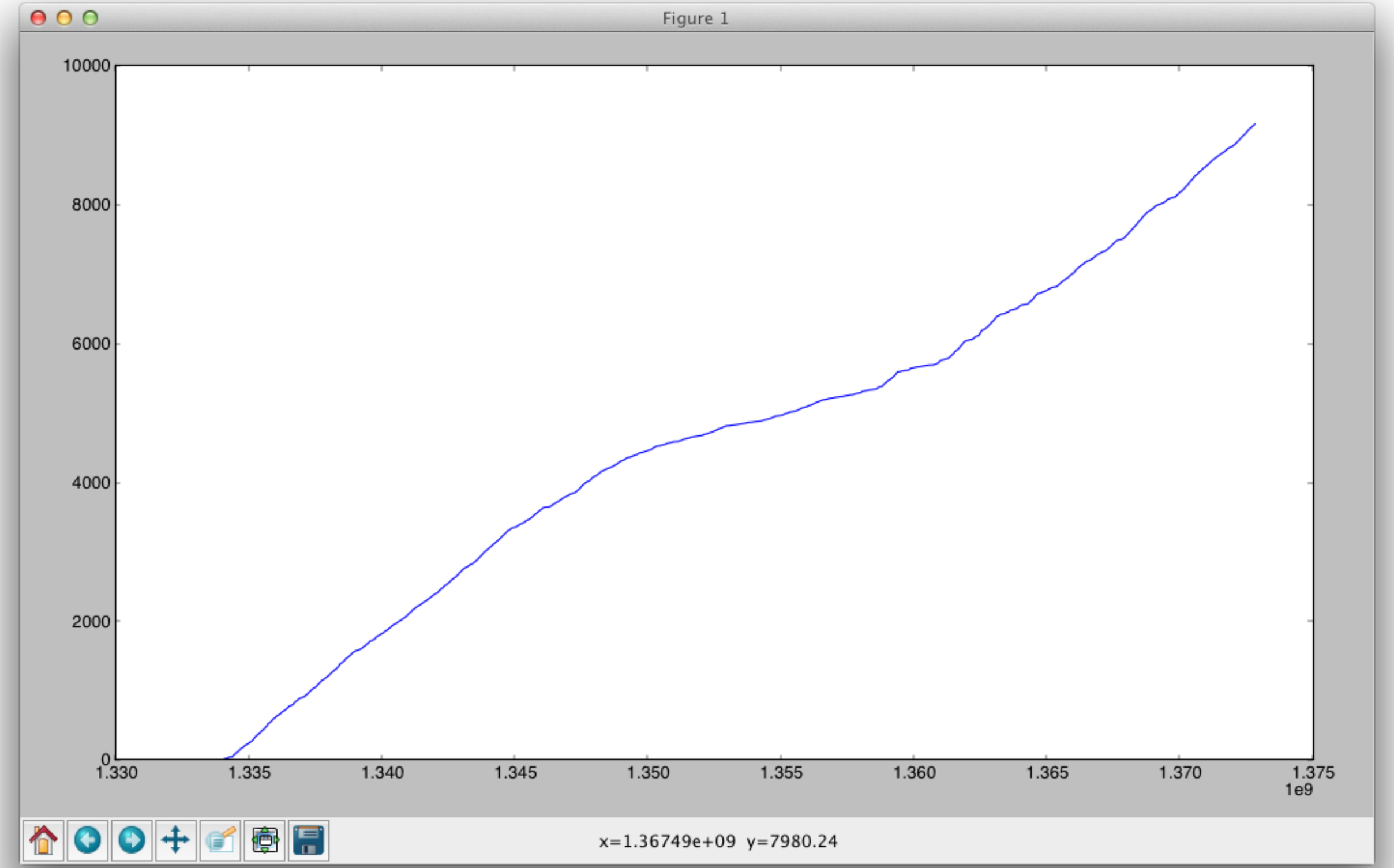
hist(temp_diff, bins=range(0,9))

Частота $\Delta T=T_i$ в соседних по времени наблюдениях, по интерв. $T_i \in \{0..8°\}$,

Но нужно получить 8 таких гистограмм,
соответствующих интервалам
в 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5 и 4.0 часа.

Кумулятивная сумма по ΔT

```
# кумулятивная сумма абс. разностей температур

>>> cum_temp = temp_diff_ip.cumsum()

>>> plot(time_range[1:], cum_temp)
```



plot(time_range[1:], cum_temp)

Дельты T в разных интервалах t

```
>>> cum_temp[1:-7] - cum_temp[:-8]  # сдвиг 1
array([1, 0, 0, ..., 0, 2, 0])

>>> cum_temp[8:]  -  cum_temp[:-8]  # сдвиг 8
array([1, 1, 1, ..., 5, 6, 5])
```

Дельты T в разных интервалах t

```
>>> cum_temp[1:-7] - cum_temp[:-8]  # сдвиг 1
array([1, 0, 0, ..., 0, 2, 0])

>>> cum_temp[8:]  -  cum_temp[:-8]  # сдвиг 8
array([1, 1, 1, ..., 5, 6, 5])

>>> temp_diffs = array([
                cum_temp[i:(-8+i) or None] - cum_temp[:-8]
                for i in range(1,9)])

>>> temp_diffs
array([[1, 0, 0, ..., 0, 2, 0],
       [1, 0, 0, ..., 2, 2, 1],
       [1, 0, 0, ..., 2, 3, 2],
       ...,
       [1, 0, 0, ..., 4, 5, 3],
       [1, 0, 1, ..., 5, 5, 4],
       [1, 1, 1, ..., 5, 6, 5]])
```

Гистограммы для каждого интервала T

```
>>> histogram(temp_diffs[0], bins=range(11))

(array([13643, 6997, 750, 104, 18, 3, 5, 2, 0, 0]),
 Array([    0,    1,   2,   3,  4, 5, 6, 7, 8, 9, 10]))

>>> histogram(temp_diffs[1], bins=range(11))

(array([9434, 7722, 3323, 804, 161, 42, 18, 8, 8, 2]),
 Array([    0,    1,   2,   3,  4,  5, 6, 7, 8, 9, 10]))

>>> histograms = array([histogram(d, bins=range(11))[0]
                        for d in temp_diffs])
```

Нормируем гистограммы

```
>>> row_sums = histograms.sum(axis=1)
>>> row_sums
array([21522, 21512, 21492, 21440, 21319, 21065, 20607, 20006])

>>> histograms = histograms.astype(float64)

# broadcasting

>>> histograms / row_sums
ValueError: operands could not be broadcast together with shapes
(8,7) (8)

>>> histograms / row_sums[:, None] # добавляем второе измерение
array([[ 0.6, 0.3, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ],
       [ 0.4, 0.4, 0.2, 0. , 0. , 0. , 0. , 0. , 0. , 0. ],
       [ 0.3, 0.3, 0.2, 0.1, 0. , 0. , 0. , 0. , 0. , 0. ],
       [ 0.3, 0.3, 0.2, 0.1, 0.1, 0. , 0. , 0. , 0. , 0. ],
       [ 0.2, 0.3, 0.2, 0.1, 0.1, 0.1, 0. , 0. , 0. , 0. ],
       [ 0.2, 0.2, 0.2, 0.2, 0.1, 0.1, 0. , 0. , 0. , 0. ],
       [ 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.1, 0. , 0. , 0. ],
       [ 0.1, 0.2, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1, 0. , 0. ]])
```

Нормируем гистограммы

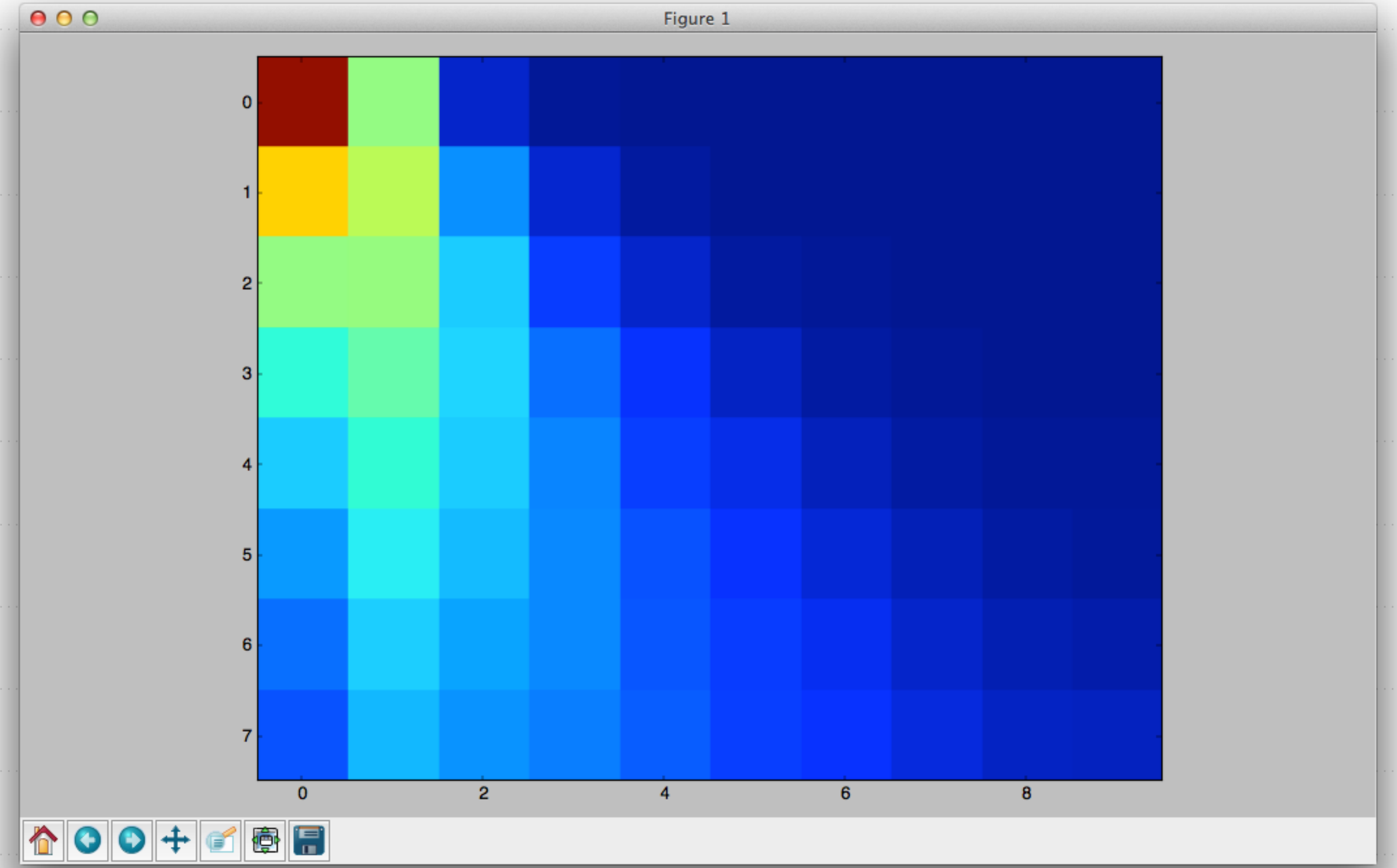
```
>>> histograms / row_sums[:, None] * 100 # векторизация!

array([[ 63.4, 32.5,  3.5,  0.5,  0.1, 0. , 0. , 0. , 0. , 0. ],
       [ 43.8, 35.9, 15.4,  3.7,  0.7, 0.2, 0.1, 0. , 0. , 0. ],
       [ 32.5, 32.9, 20. ,  9.6,  3.5, 1. , 0.3, 0.2, 0.1, 0.1],
       [ 25.1, 29. , 20.7, 13.2,  7. , 3.1, 1.1, 0.4, 0.2, 0.2],
       [ 20. , 25.5, 20. , 14.7,  9.7, 5.6, 2.5, 1.1, 0.5, 0.4],
       [ 16.3, 22.6, 18.6, 15. , 11.3, 7.6, 4.4, 2.3, 1.1, 0.7],
       [ 13.4, 20.3, 17.1, 15. , 11.5, 9.5, 6. , 3.6, 2.1, 1.6],
       [ 11.2, 18.4, 15.8, 14.1, 12. , 9.9, 7.7, 4.9, 3.1, 2.9]])

>>> histogram_percents = _
```

Нормируем гистограммы

```
>>> histograms / row_sums[:, None] * 100 # векторизация!

array([[ 63.4, 32.5,  3.5,  0.5,  0.1, 0. , 0. , 0. , 0. , 0. ],
       [ 43.8, 35.9, 15.4,  3.7,  0.7, 0.2, 0.1, 0. , 0. , 0. ],
       [ 32.5, 32.9, 20. ,  9.6,  3.5, 1. , 0.3, 0.2, 0.1, 0.1],
       [ 25.1, 29. , 20.7, 13.2,  7. , 3.1, 1.1, 0.4, 0.2, 0.2],
       [ 20. , 25.5, 20. , 14.7,  9.7, 5.6, 2.5, 1.1, 0.5, 0.4],
       [ 16.3, 22.6, 18.6, 15. , 11.3, 7.6, 4.4, 2.3, 1.1, 0.7],
       [ 13.4, 20.3, 17.1, 15. , 11.5, 9.5, 6. , 3.6, 2.1, 1.6],
       [ 11.2, 18.4, 15.8, 14.1, 12. , 9.9, 7.7, 4.9, 3.1, 2.9]])

>>> histogram_percents = _

>>> histogram_percents.sum(axis=1)
Array([ 100.,  100.,  100.,  100.,  100.,  100.,  100.,  100.])
```

Нормируем гистограммы

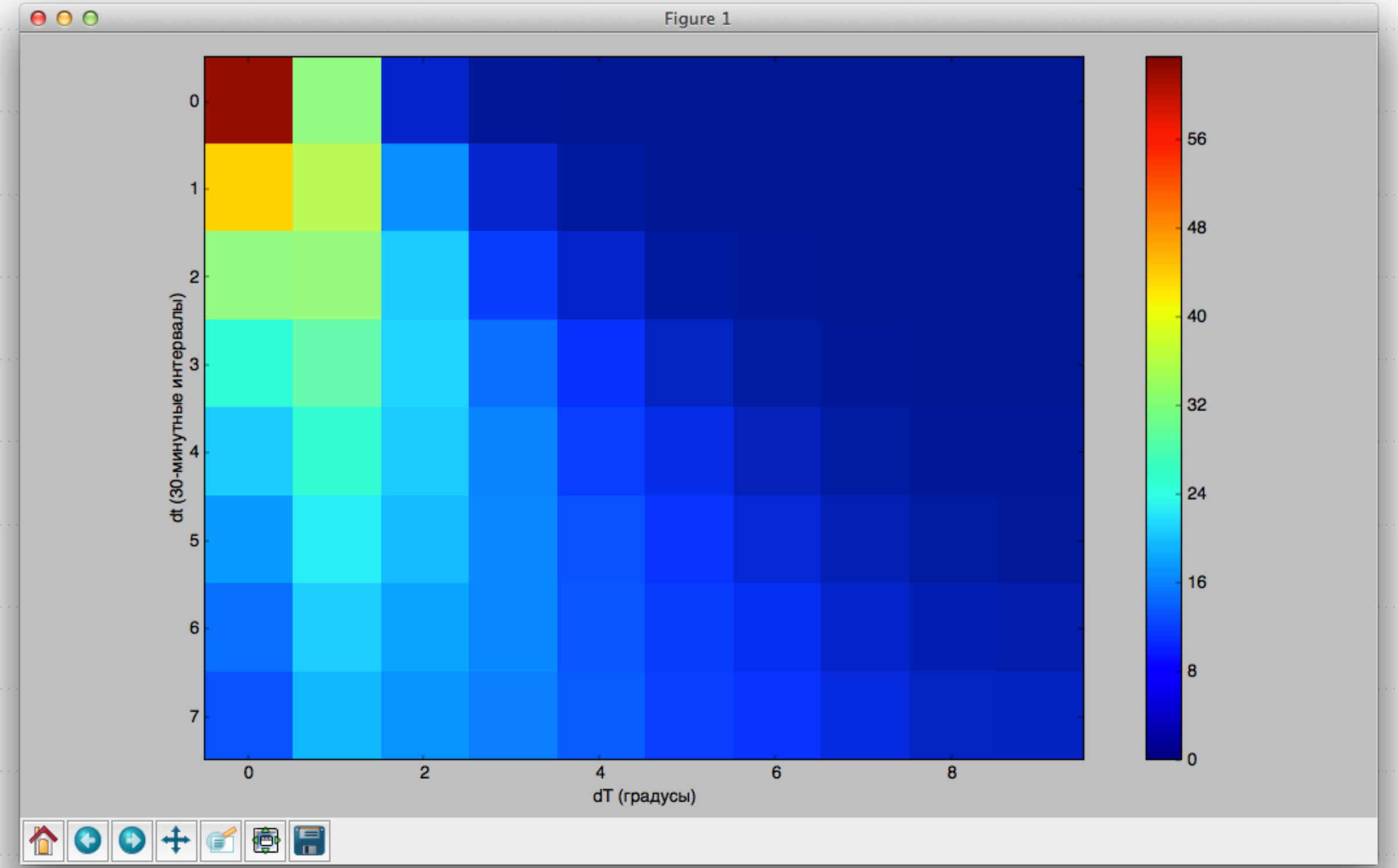
```
>>> histograms / row_sums[:, None] * 100 # векторизация!

array([[ 63.4, 32.5,  3.5,  0.5,  0.1, 0. , 0. , 0. , 0. , 0. ],
       [ 43.8, 35.9, 15.4,  3.7,  0.7, 0.2, 0.1, 0. , 0. , 0. ],
       [ 32.5, 32.9, 20. ,  9.6,  3.5, 1. , 0.3, 0.2, 0.1, 0.1],
       [ 25.1, 29. , 20.7, 13.2,  7. , 3.1, 1.1, 0.4, 0.2, 0.2],
       [ 20. , 25.5, 20. , 14.7,  9.7, 5.6, 2.5, 1.1, 0.5, 0.4],
       [ 16.3, 22.6, 18.6, 15. , 11.3, 7.6, 4.4, 2.3, 1.1, 0.7],
       [ 13.4, 20.3, 17.1, 15. , 11.5, 9.5, 6. , 3.6, 2.1, 1.6],
       [ 11.2, 18.4, 15.8, 14.1, 12. , 9.9, 7.7, 4.9, 3.1, 2.9]])

>>> histogram_percents = _

>>> histogram_percents.sum(axis=1)
Array([ 100.,  100.,  100.,  100.,  100.,  100.,  100.,  100.])

>>> imshow(histogram_percents, interpolation="nearest")
```



imshow(histogram_percents, interpolation="nearest")
Частота попадания ΔT в интервалы 0..10°, по вр. интервалам

Наводим красоту

```
>>> imshow(histogram_percents, interpolation="nearest")
>>> ylabel(u'dt (30-минутные интервалы)')
>>> xlabel(u'dT (градусы)')
>>> colorbar()
```



imshow(histogram_percents, interpolation="nearest")

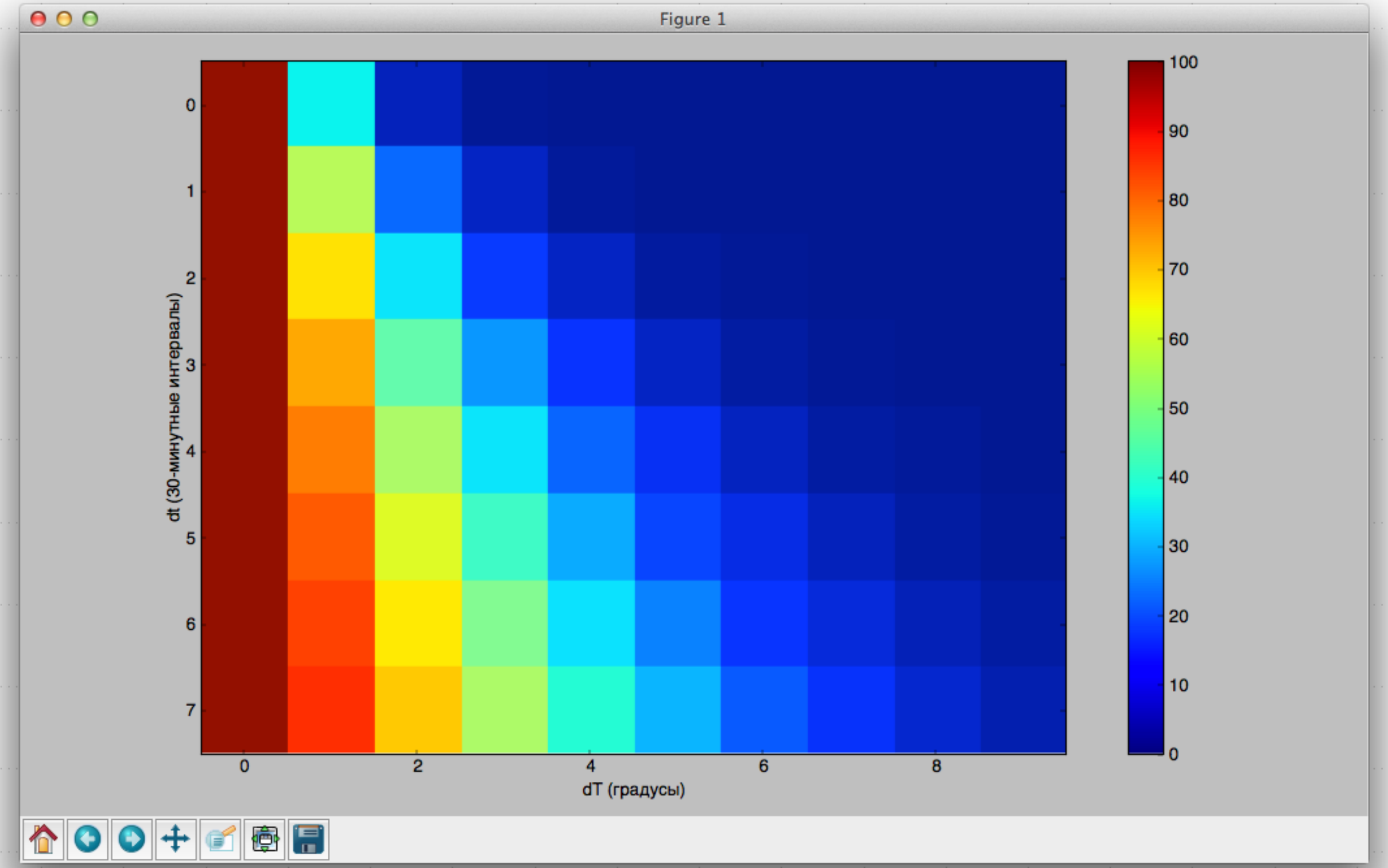
Частота попадания ΔT в интервалы 0..10°, по вр. интервалам

Кумулятивная гистограмма

```
>>> histogram_percents.cumsum(axis=1)

#      ΔT<=0   <=1   <=2   <=3   <=4   <=5   <=6   <=7   <=8   --
array([[63.4, 95.9, 99.4, 99.9,100. ,100. ,100. ,100. ,100. , 100.],
       [43.8, 79.7, 95.2, 98.9, 99.6, 99.8, 99.9,100. ,100. , 100.],
       [32.5, 65.3, 85.3, 94.9, 98.5, 99.5, 99.7, 99.9, 99.9, 100.],
       [25.1, 54.1, 74.9, 88.1, 95.1, 98.2, 99.3, 99.7, 99.8, 100.],
       [20. , 45.5, 65.5, 80.2, 89.9, 95.5, 98.1, 99.2, 99.6, 100.],
       [16.3, 38.8, 57.5, 72.5, 83.8, 91.5, 95.9, 98.2, 99.3, 100.],
       [13.4, 33.7, 50.7, 65.7, 77.2, 86.7, 92.7, 96.3, 98.4, 100.],
       [11.2, 29.5, 45.3, 59.5, 71.5, 81.3, 89.1, 94. , 97.1, 100.]])
```

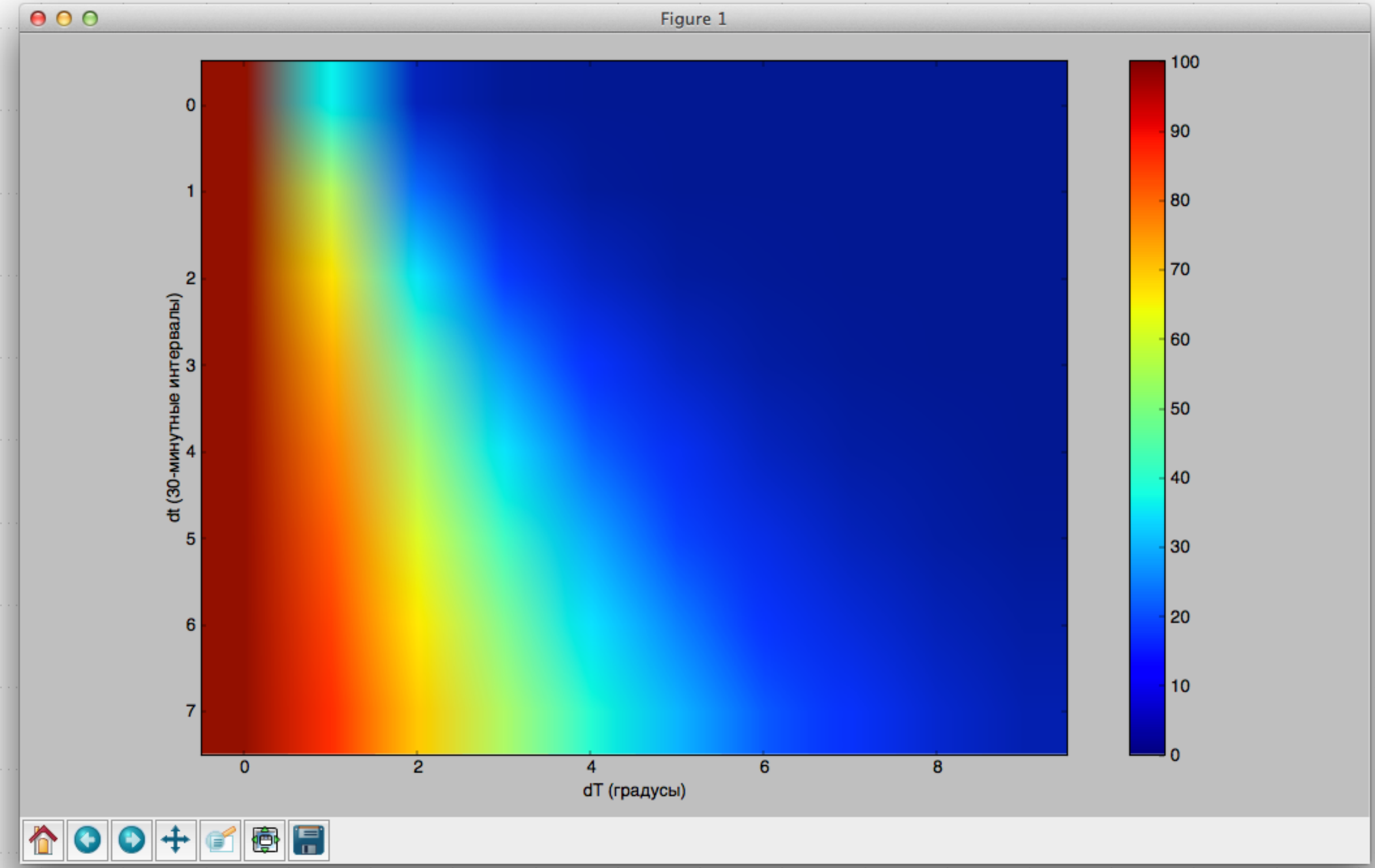
Кумулятивная гистограмма

```
#   отзеркалить, просуммировать кумулятивно, отзеркалить
>>> histogram_percents[:, ::-1].cumsum(axis=1)[:, ::-1]

Array([
#ΔT≥0   ≥1    ≥2     ≥3    ≥4    ≥5     ≥6   ≥7   ≥8   ≥9°
[100.,  36.6,   4.1,   0.6,   0.1,   0. ,   0. ,  0. , 0. , 0. ], # 0.5ч
[100.,  56.2,  20.3,   4.8,   1.1,   0.4,   0.2,  0.1, 0. , 0. ], # 1.0ч
[100.,  67.5,  34.7,  14.7,   5.1,   1.5,   0.5,  0.3, 0.1, 0.1], # 1.5ч
[100.,  74.9,  45.9,  25.1,  11.9,   4.9,   1.8,  0.7, 0.3, 0.2], # 2.0ч
[100.,  80. ,  54.5,  34.5,  19.8,  10.1,   4.5,  1.9, 0.8, 0.4], # 2.5ч
[100.,  83.7,  61.2,  42.5,  27.5,  16.2,   8.5,  4.1, 1.8, 0.7], # 3.0ч
[100.,  86.6,  66.3,  49.3,  34.3,  22.8,  13.3,  7.3, 3.7, 1.6], # 3.5ч
[100.,  88.8,  70.5,  54.7,  40.5,  28.5,  18.7, 10.9, 6. , 2.9]  # 4.0ч
])
>>> histogram_cum = _
>>> savetxt('histogram_cum.txt', histogram_cum)
```



imshow(histogram_cum, interpolation="nearest")

Частота того, что $\Delta_t T \le T_i$, $T_i \in \{0..10°\}$, $\Delta_t \in \{0.5..4.0ч\}$



imshow(histogram_cum)

Частота того, что $\Delta_t T \le T_i$, $T_i \in \{0..10°\}$, $\Delta_t \in \{0.5..4.0ч\}$

Чем больше времени проходит, тем сильнее “размывается” температура

Где ответ?

```
# Через какое время T изменится на 2° и более,
# с вероятностью 50%?

#ΔT≥0  ≥1    ≥2    ≥3   ≥4   ≥5    ≥6   ≥7   ≥8   ≥9°

[100., 36.6,  4.1,  0.6,  0.1,  0. ,  0. ,  0. , 0. , 0. ], # 0.5ч
[100., 56.2, 20.3,  4.8,  1.1,  0.4,  0.2,  0.1, 0. , 0. ], # 1.0ч
[100., 67.5, 34.7, 14.7,  5.1,  1.5,  0.5,  0.3, 0.1, 0.1], # 1.5ч
[100., 74.9, 45.9, 25.1, 11.9,  4.9,  1.8,  0.7, 0.3, 0.2], # 2.0ч
[100., 80. , 54.5, 34.5, 19.8, 10.1,  4.5,  1.9, 0.8, 0.4], # 2.5ч
[100., 83.7, 61.2, 42.5, 27.5, 16.2,  8.5,  4.1, 1.8, 0.7], # 3.0ч
[100., 86.6, 66.3, 49.3, 34.3, 22.8, 13.3,  7.3, 3.7, 1.6], # 3.5ч
[100., 88.8, 70.5, 54.7, 40.5, 28.5, 18.7, 10.9, 6. , 2.9]  # 4.0ч
```

После Δt=2.5ч температура меняется на 2°C и более, с вероятностью 50%

Ссылки

- numpy.org
- matplotlib.org
- habrahabr.ru/search/?q=numpy
- scipy.org
- conference.scipy.org/scipy2013/
- github.com/fonnesbeck/statistical-analysis-python-tutorial

Яндекс

Руслан Гроховецкий

Руководитель группы
справочных сервисов

ruguevara@yandex-team.ru

Спасибо