Visualizing time series

INTRODUCTION TO DATA VISUALIZATION IN PYTHON

Bryan Van de Ven
Core Developer of Bokeh

Datetimes & time series

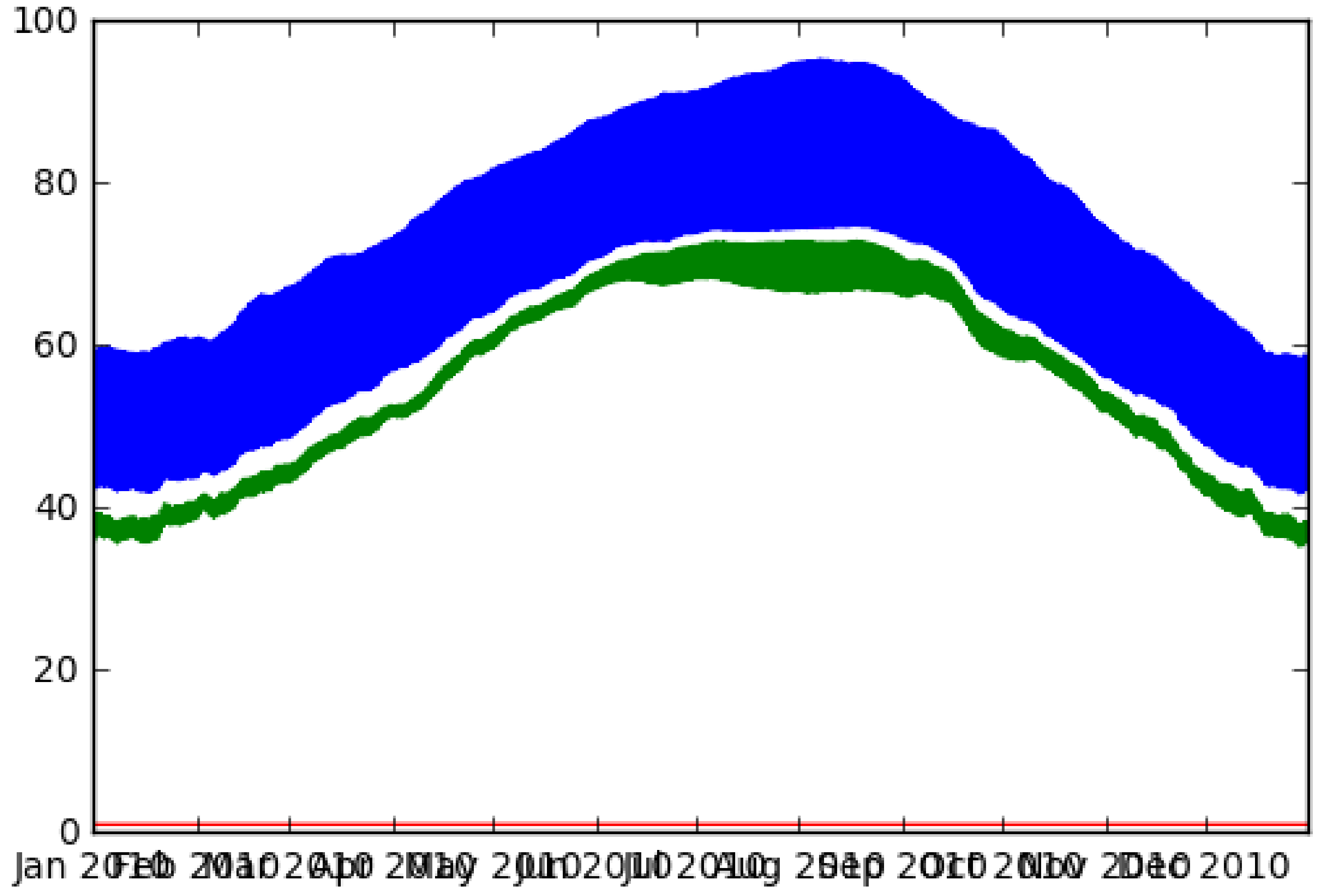
```
type(weather), type(weather.index)
```

```
(pandas.core.frame.DataFrame, pandas.tseries.index.DatetimeIndex)
```

Date	Temperature	DewPoint	Pressure
2010-01-01 00:00:00 AM	46.2	37.5	1
2010-01-01 01:00:00 AM	44.6	37.1	1
2010-01-01 02:00:00 AM	44.1	36.9	1
...	...	...	...

Plotting DataFrames

```
plt.plot(weather)
plt.show()
```



Time series

- `pandas` time series: `datetime` as index
- Datetime: represents periods or time-stamps
- Datetime index: specialized slicing
  - `weather['2010-07-04']`
  - `weather['2010-03':'2010-04']`
  - `weather['2010-05']`
  - etc.

Slicing time series

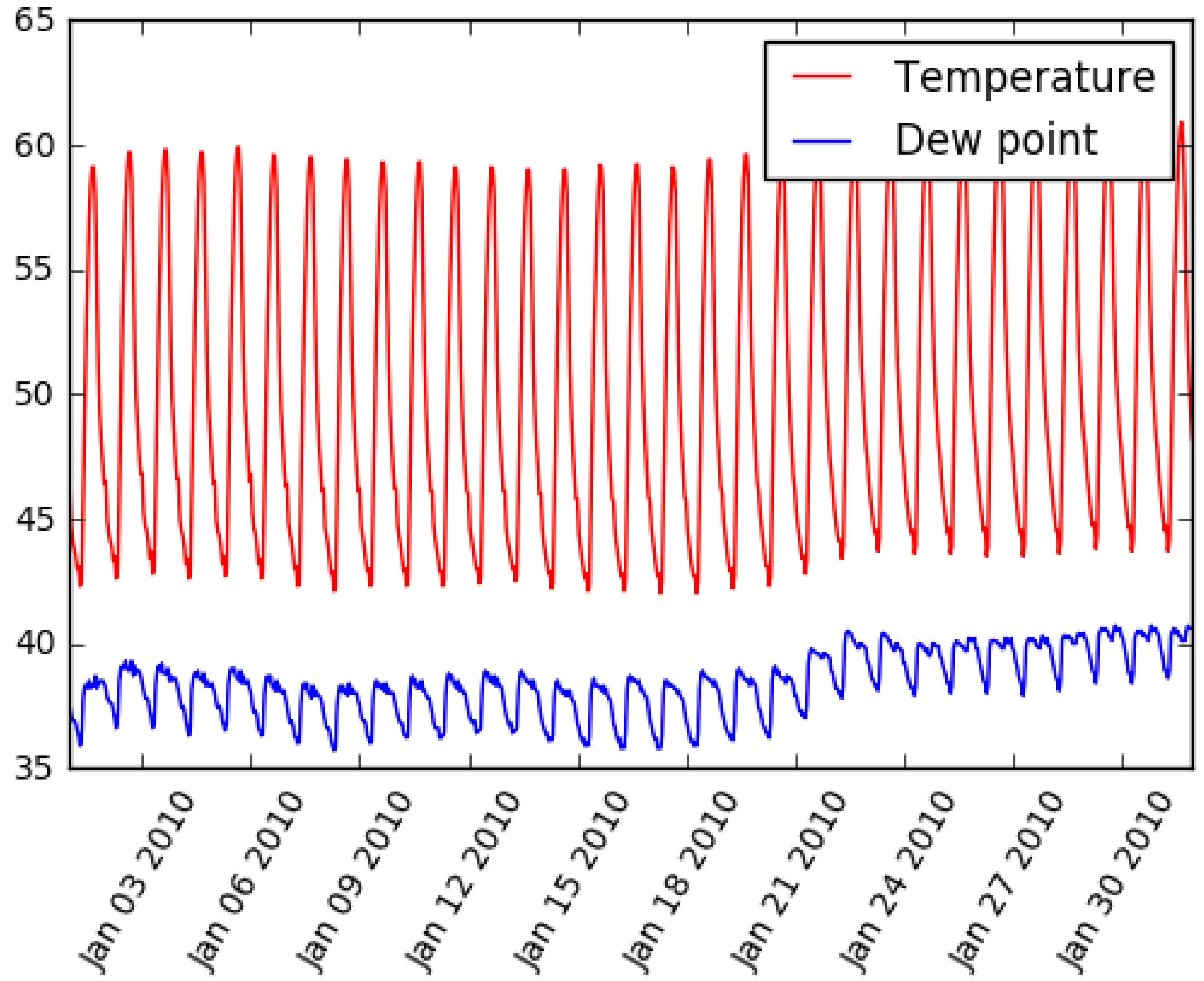
```
temperature = weather['Temperature']
march_apr = temperature['2010-03':'2010-04']  # data of March & April 2010 only
march_apr.shape
```

```
(1463,)
```

```
march_apr.iloc[-4:] #extract last 4 entries from time series
```

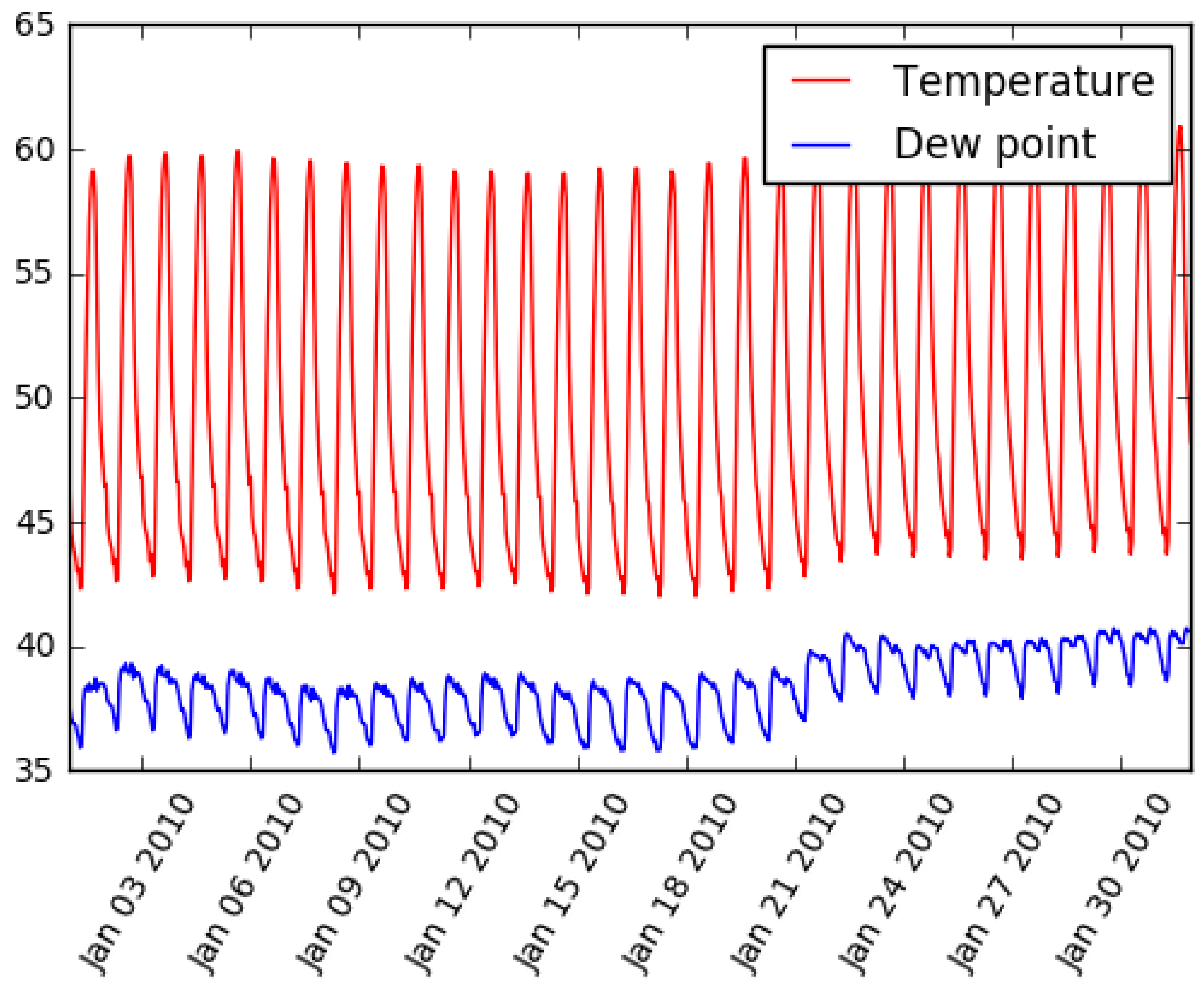
```
Date
2010-04-30 20:00:00    73.3
2010-04-30 21:00:00    71.3
2010-04-30 22:00:00    69.7
2010-04-30 23:00:00    68.5
Name: Temperature, dtype: float64
```

Plotting time series slices



Plotting time series slices

```
plt.plot(temperature['2010-01'],
             color='red',
             label='Temperature')

dew point = weather['DewPoint']
plt.plot(dewpoint['2010-01'],
             color='blue',
             label='Dewpoint')

plt.legend(loc='upper right')
plt.xticks(rotation=60)
plt.show()
```



Selecting & formatting dates

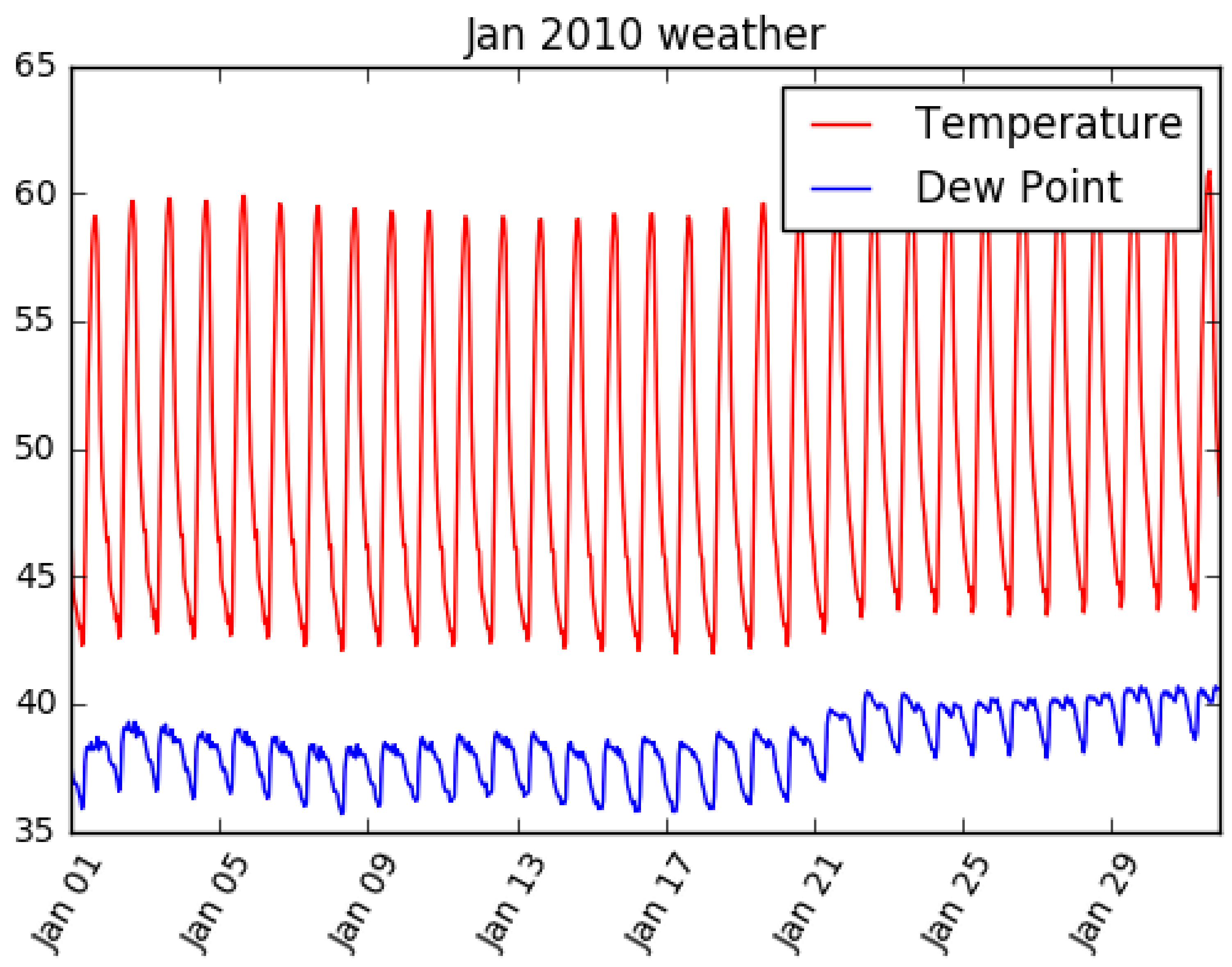
```
jan = temperature['2010-01']
dates = jan.index[::96]  # Pick every 4th day
print(dates)
```

```
DatetimeIndex(['2010-01-01', '2010-01-05', '2010-01-09', '2010-01-13','2010-01-17',
'2010-01-21', '2010-01-25', '2010-01-29'], dtype='datetime64[ns]', name='Date', freq=None)
```

```
labels = dates.strftime('%b %d') # Make formatted labels
print(labels)
```

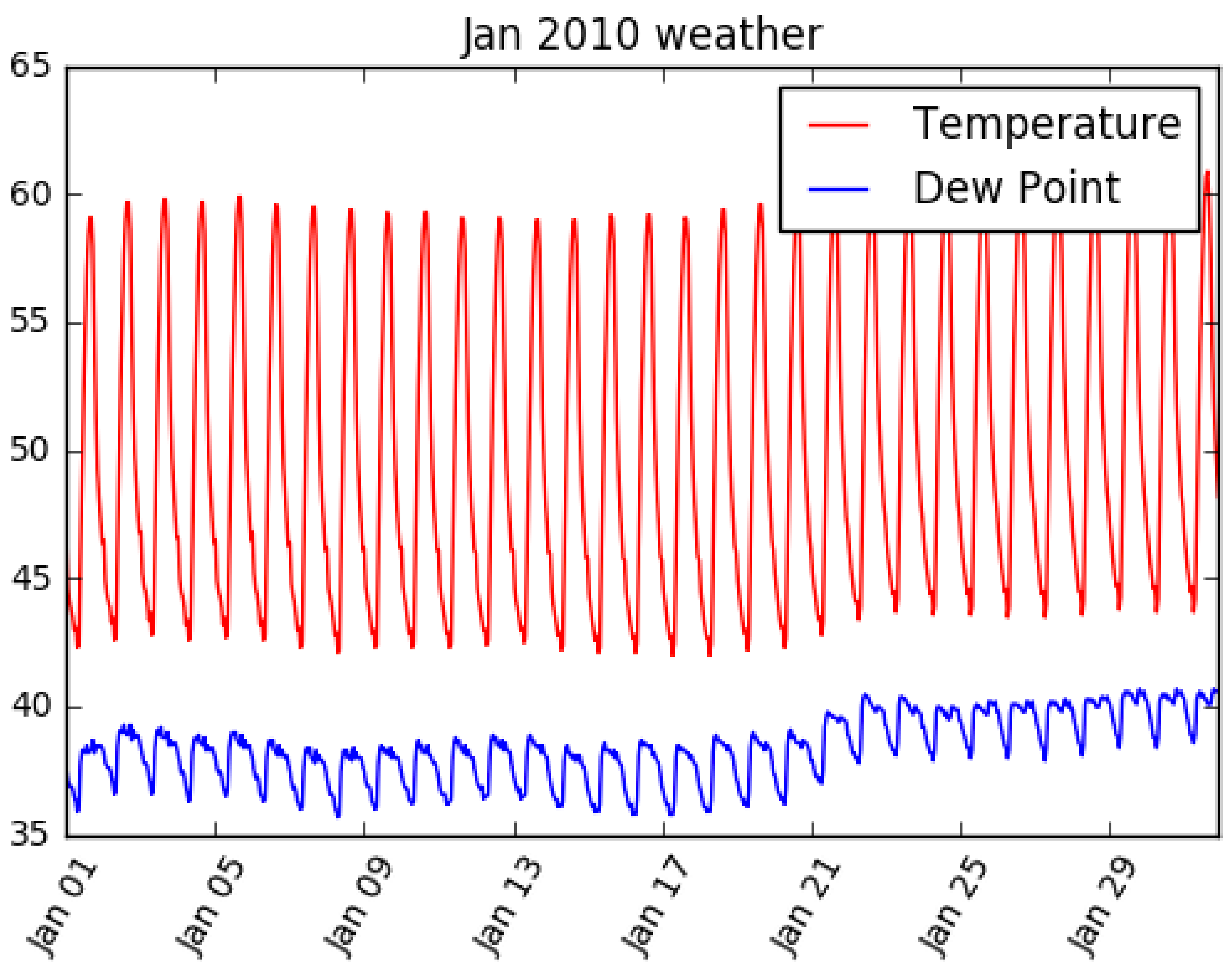
```
['Jan 01' 'Jan 05' 'Jan 09' 'Jan 13' 'Jan 17' 'Jan 21' 'Jan 25' 'Jan 29']
```

Cleaning up ticks on axis



Cleaning up ticks on axis

```
plt.plot(temperature['2010-01'],
             color='red',
             label='Temperature')
plt.plot(dewpoint['2010-01'],
             color='blue',
             label='Dewpoint')

plt.xticks(dates, labels, rotation=60)

plt.legend(loc='upper right')
plt.show()
```



Let's practice!

INTRODUCTION TO DATA VISUALIZATION IN PYTHON

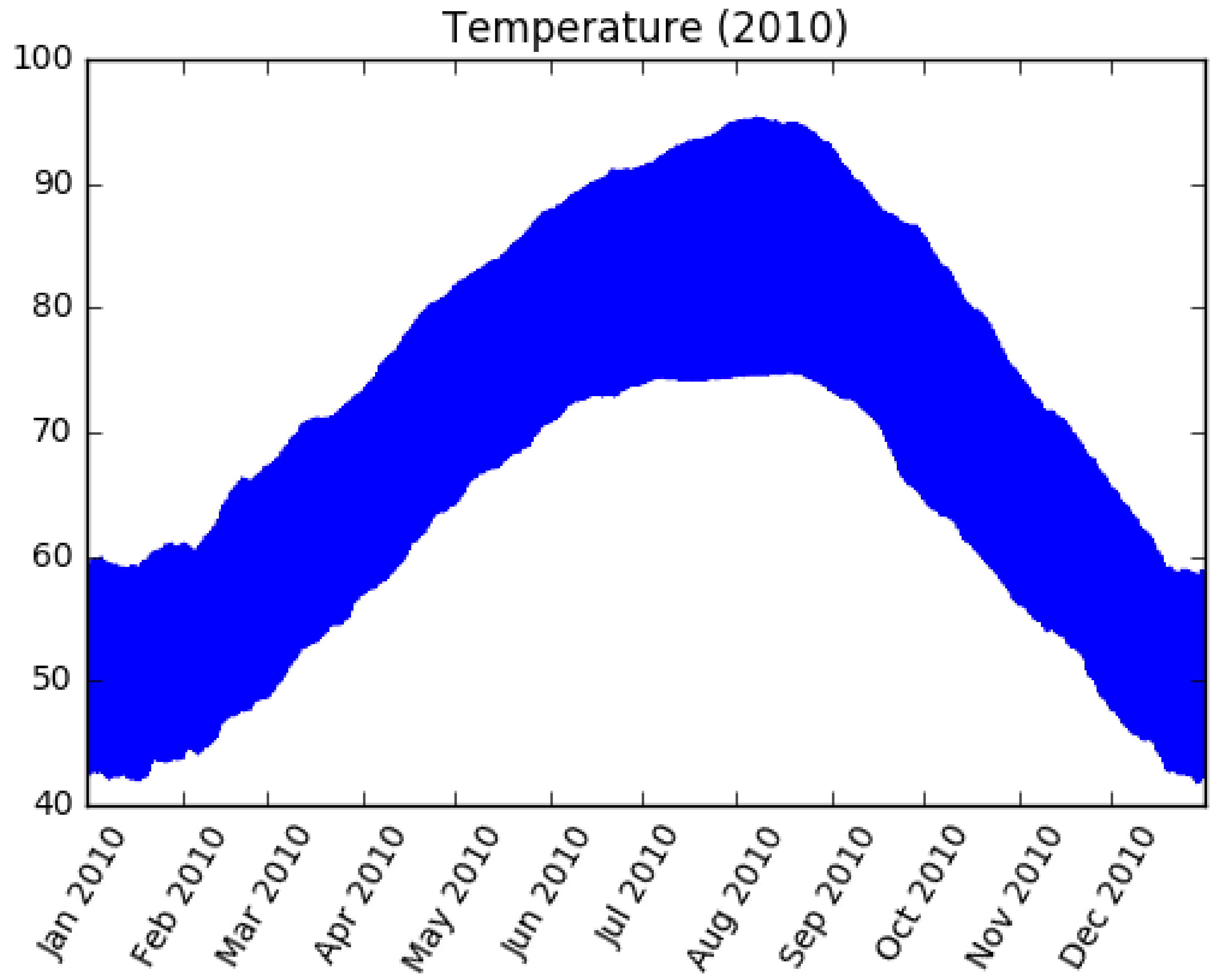
Time series with moving windows

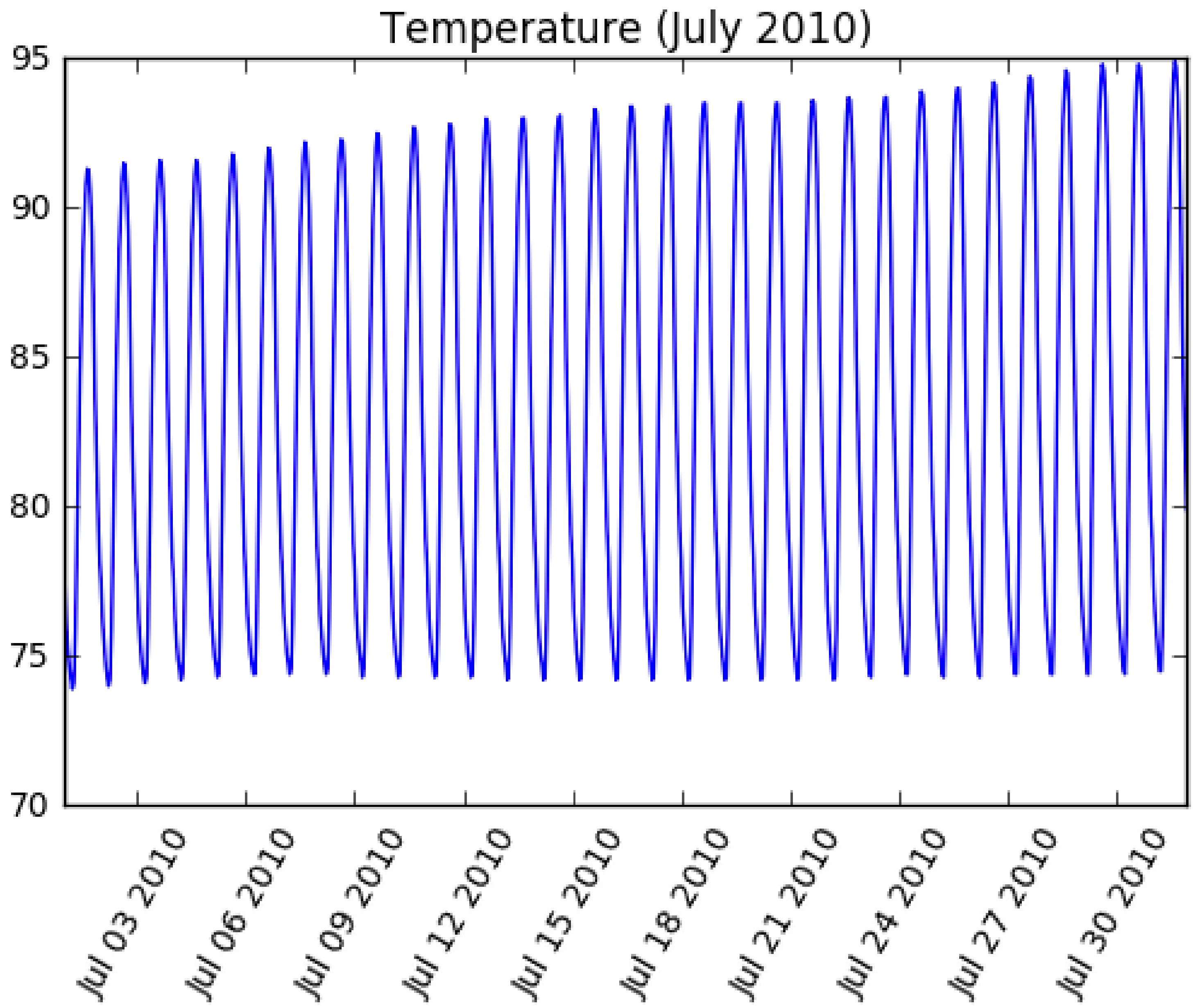
INTRODUCTION TO DATA VISUALIZATION IN PYTHON

Bryan Van de Ven
Core Developer of Bokeh

datacamp

Hourly data over a year





Moving windows and time series

- Moving window calculations
  - Averages
  - Medians
  - Standard deviations
- Extracts information on longer time scales
- See `pandas` courses on how to compute

Moving averages

```python
# smoothed computing using moving averages
smoothed.info()
```

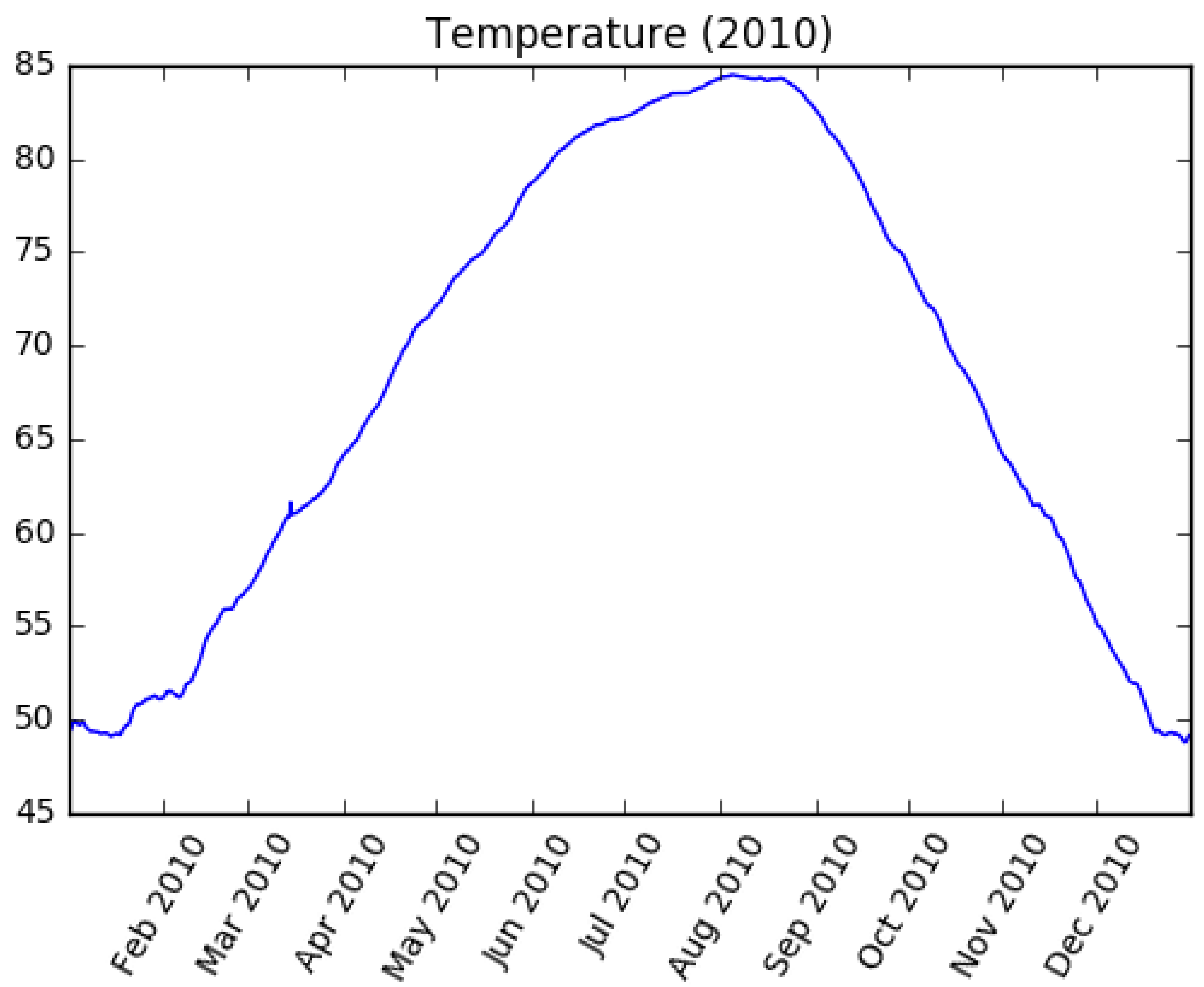
```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 8759 entries,
2010-01-01 00:00:00 to 2010-12-31 23:00:00
Data columns (total 5 columns):
14d              8424 non-null float64
1d               8736 non-null float64
3d               8688 non-null float64
7d               8592 non-null float64
dtypes: float64(5)
memory usage: 410.6 KB
```

```python
print(smoothed.iloc[:3,:])
```

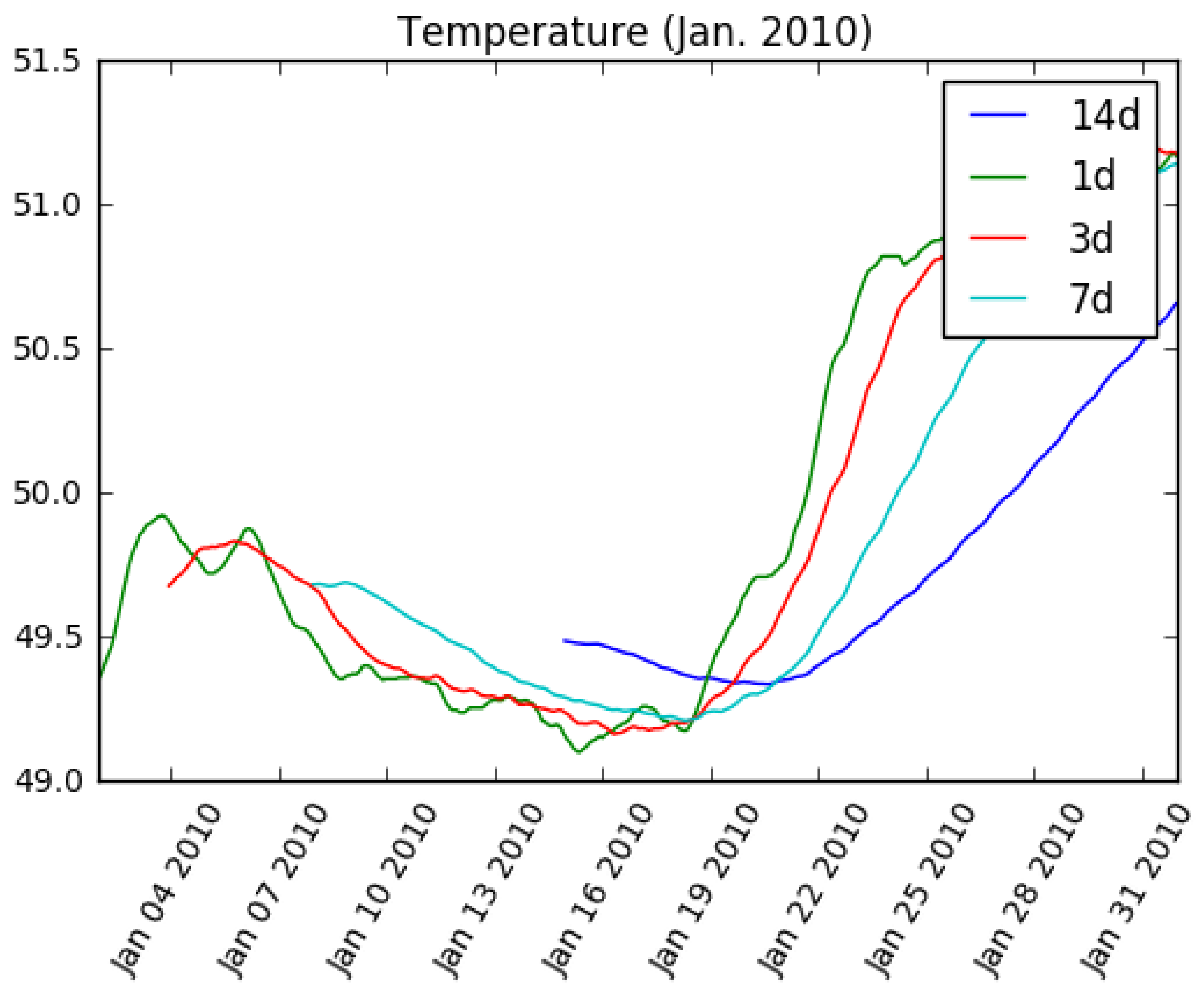
```
                     14d  1d  3d  7d  Temperature
Date
2010-01-01 00:00:00  NaN NaN NaN NaN         46.2
2010-01-01 01:00:00  NaN NaN NaN NaN         44.6
2010-01-01 02:00:00  NaN NaN NaN NaN         44.1
```

Viewing 24 hour averages

```
# moving average over 24 hours
plt.plot(smoothed['1d'])
plt.title('Temperature (2010)')
plt.xticks(rotation=60)
plt.show()
```

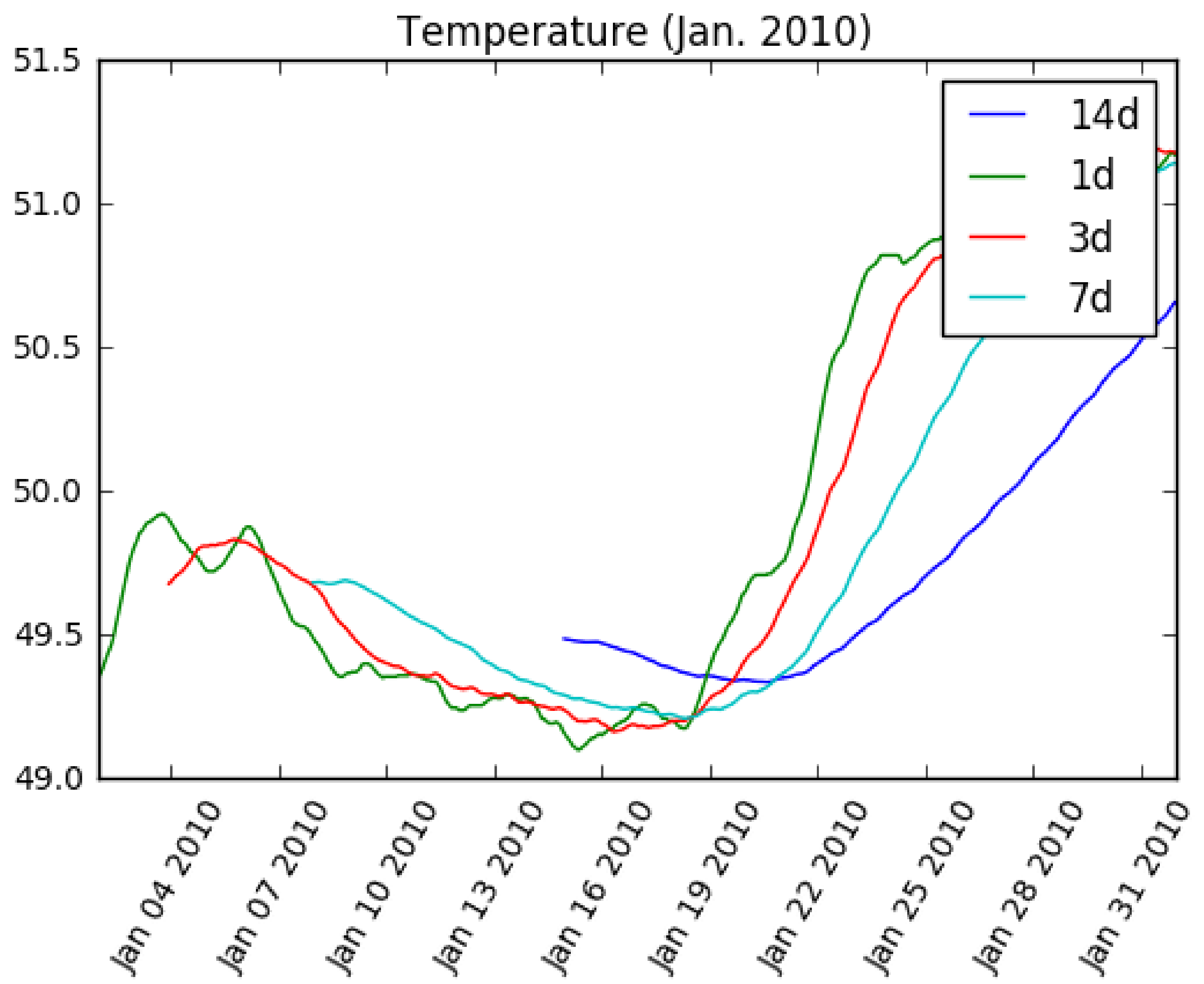


Viewing all moving averages

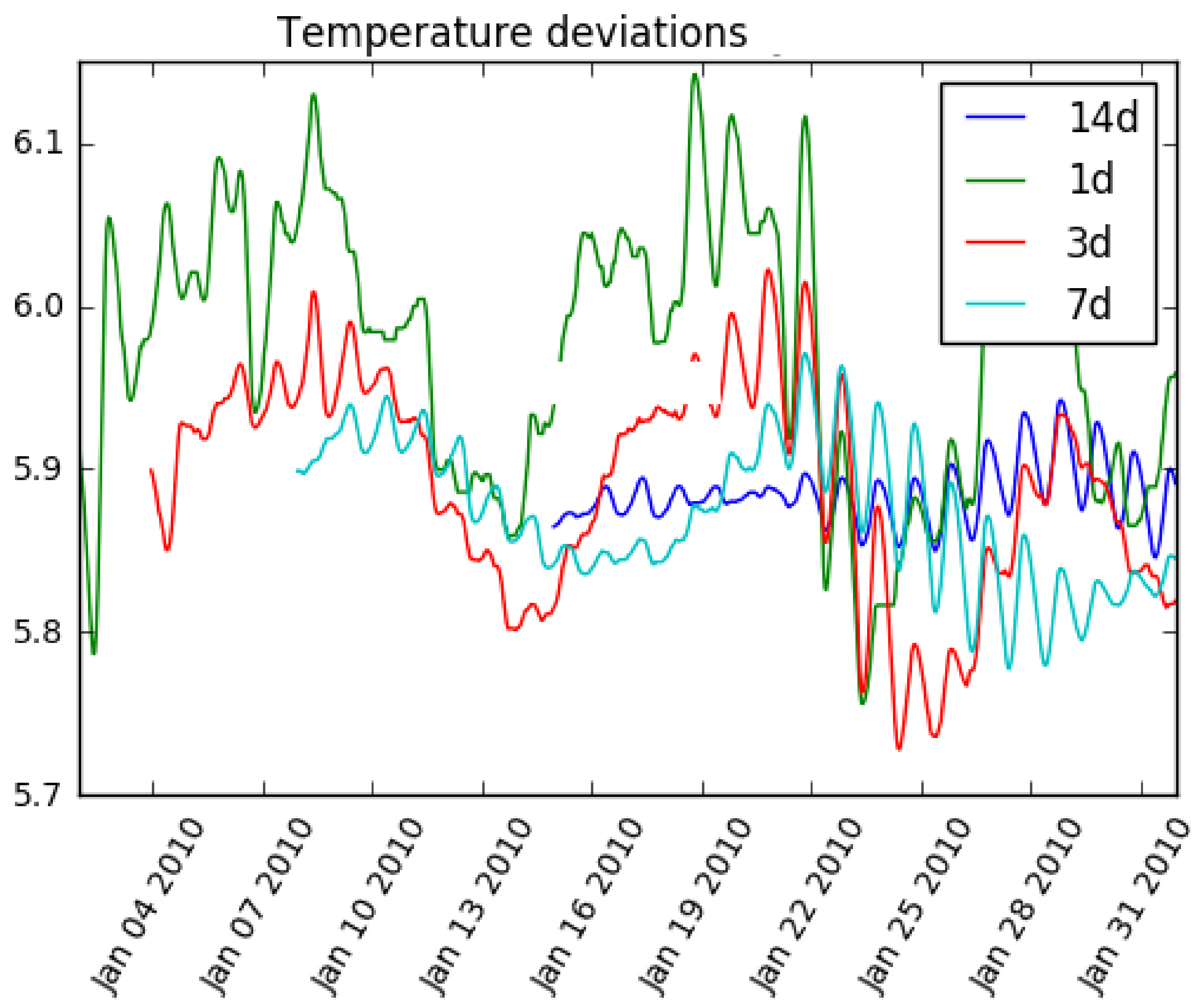


Viewing all moving averages

```
# plot DataFrame for January
plt.plot(smoothed['2010-01'])
plt.legend(smoothed.columns)
plt.title('Temperature (Jan. 2010)')
plt.xticks(rotation=60)
plt.show()
```

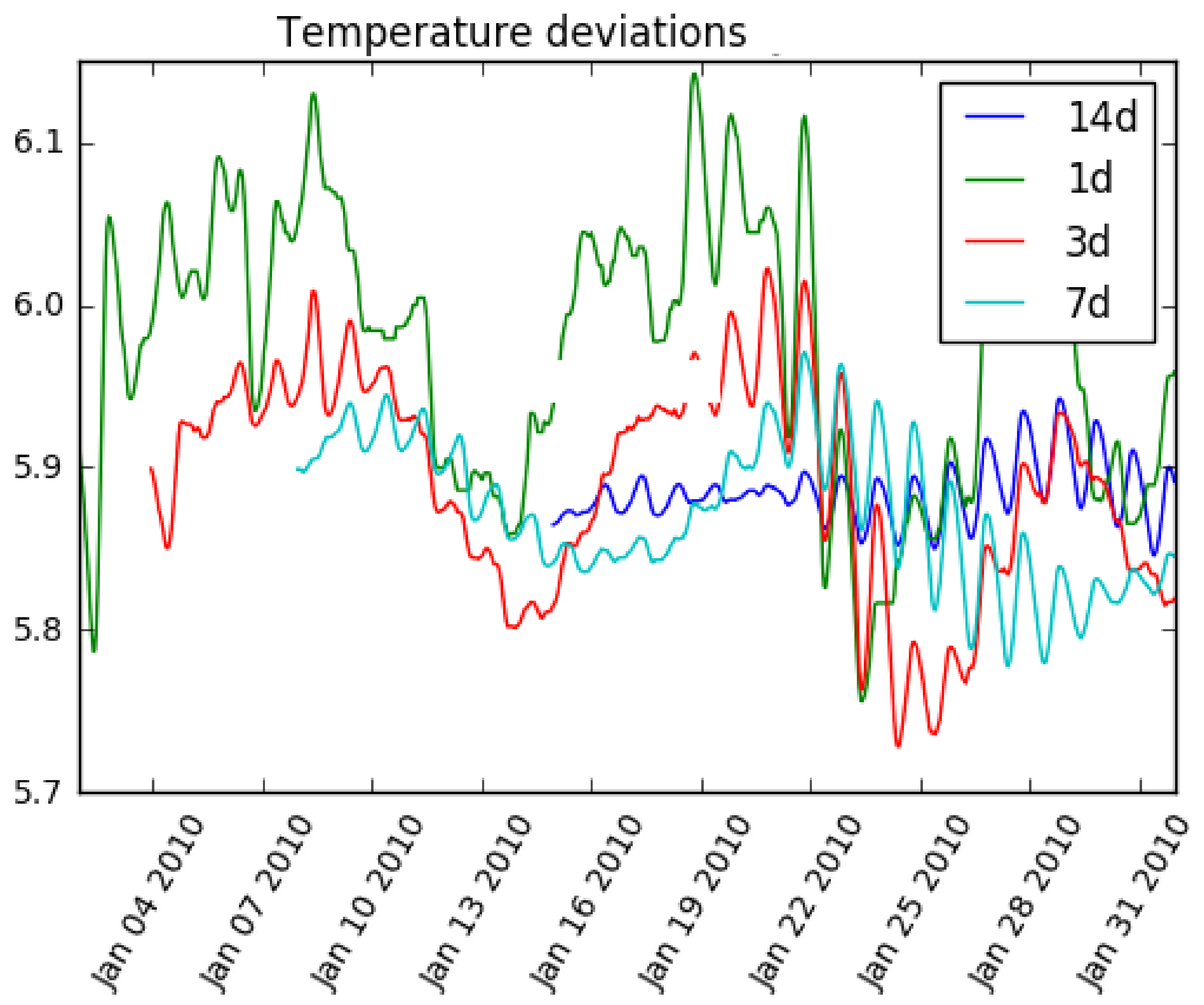


Moving standard deviations



Moving standard deviations

```
plt.plot(variances['2010-01'])
plt.legend(variances.columns)
plt.title('Temperature deviations')
plt.xticks(rotation=60)
plt.show()
```



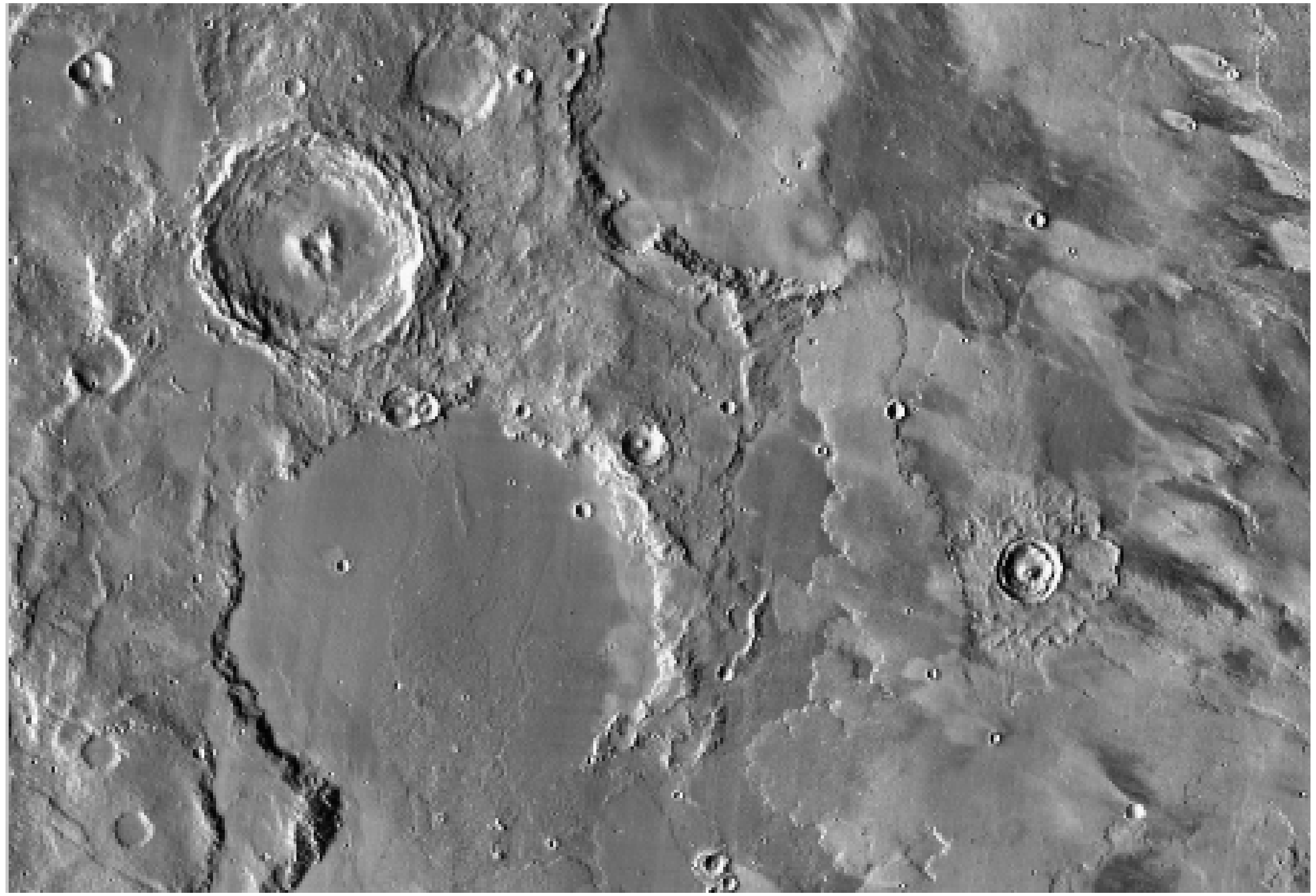
Let's practice!

Histogram equalization in images

INTRODUCTION TO DATA VISUALIZATION IN PYTHON

Bryan Van de Ven
Core Developer of Bokeh

Original image



Equalized image

Equalized image

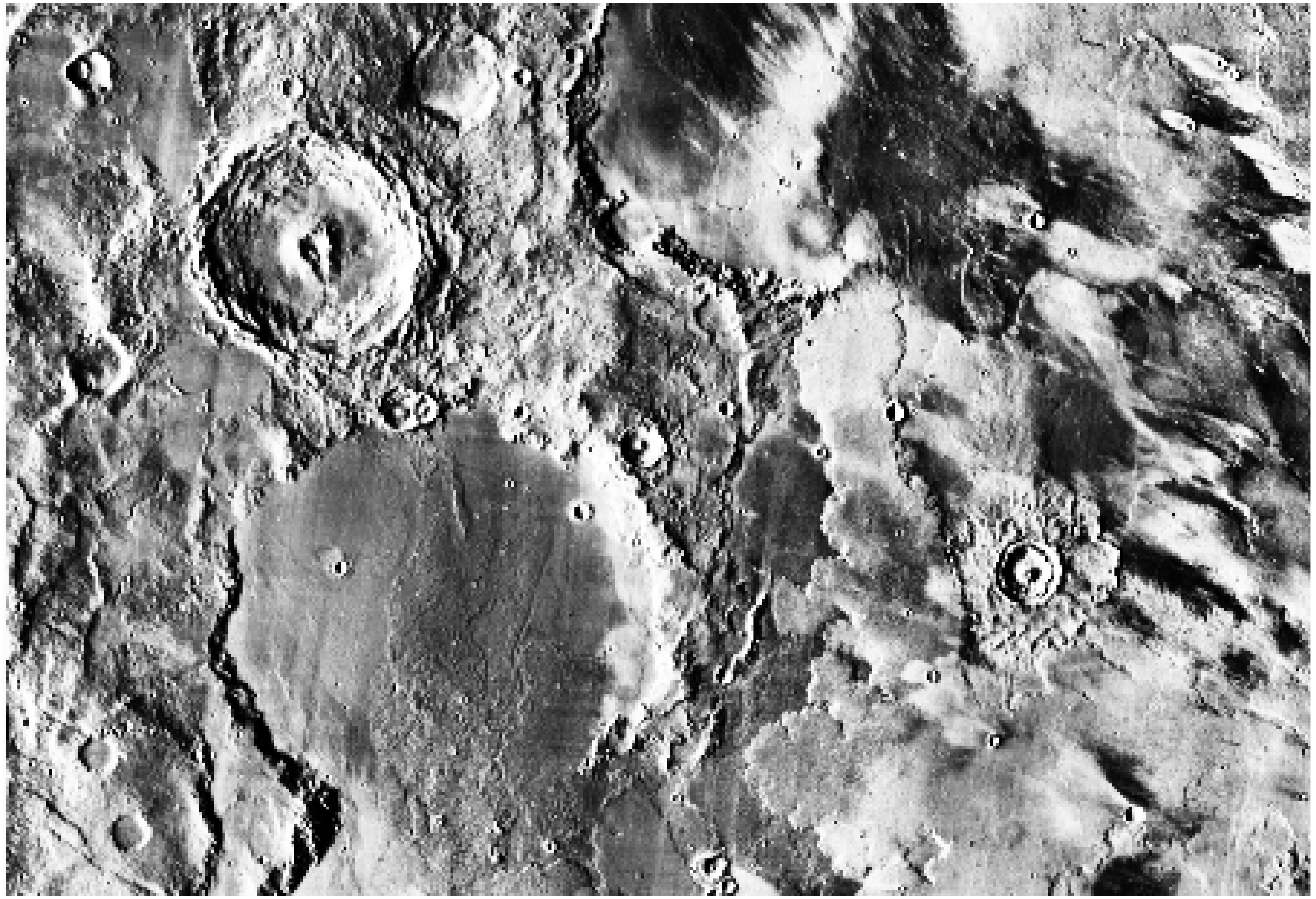


Image histograms

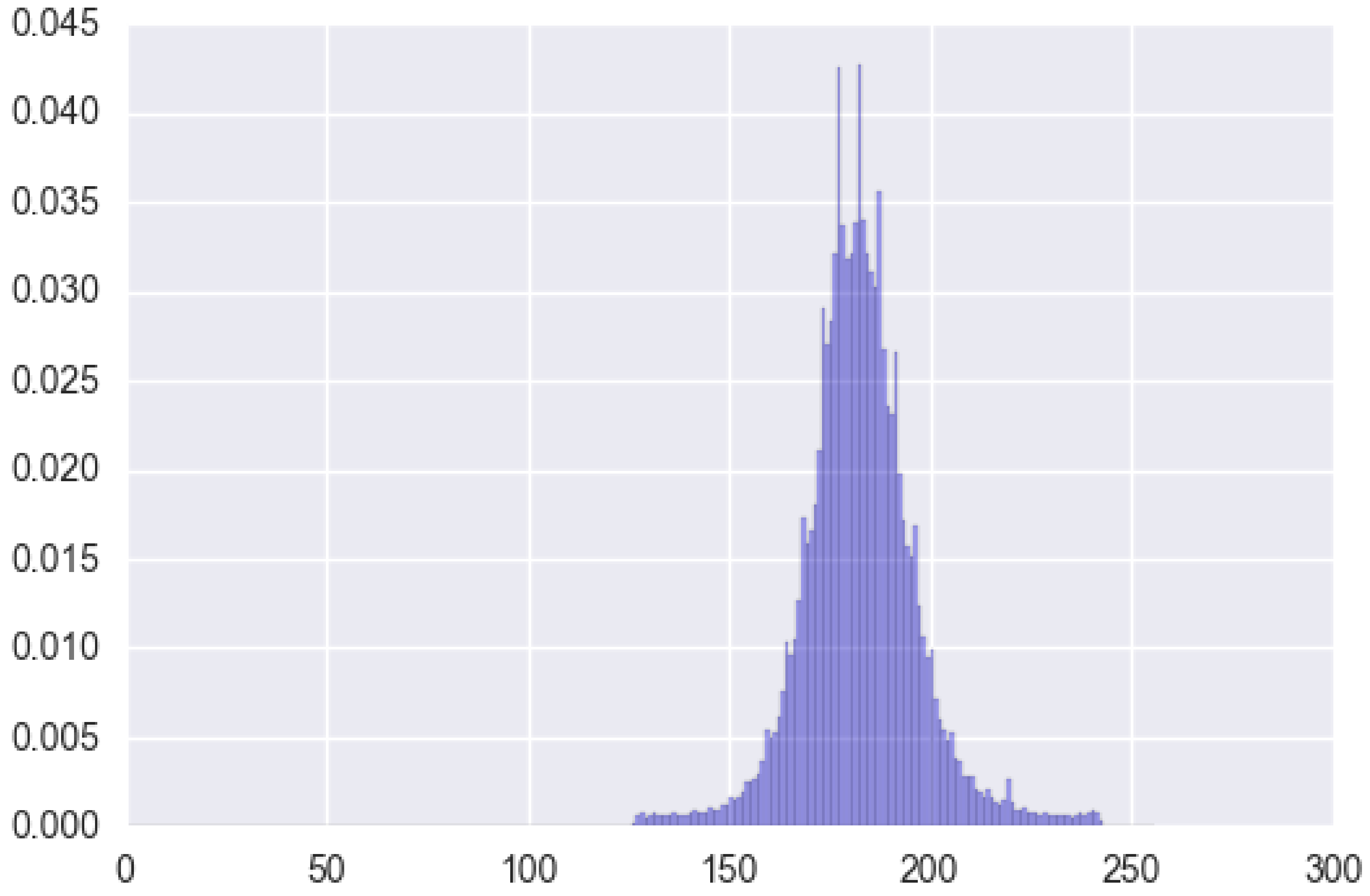
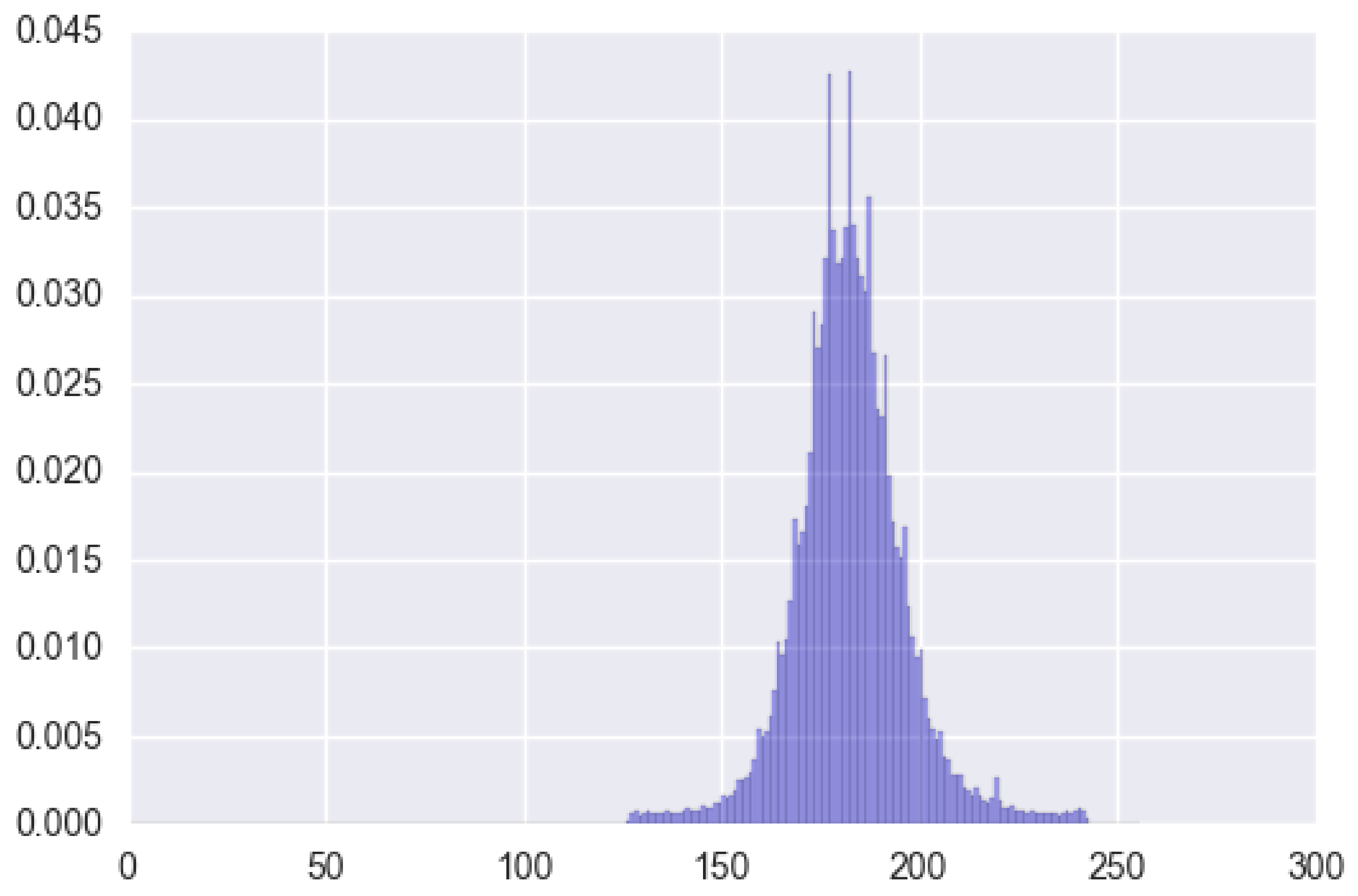


Image histograms

```
orig = plt.imread('low-contrast-moon.jpg')
pixels = orig.flatten()
plt.hist(pixels, bins=256, range=(0,256),
                 normed=True,
                 color='blue', alpha=0.3)
plt.show()
minval, maxval = orig.min(), orig.max()
print(minval, maxval)
```

```
125 244
```



Rescaling the image

```
minval, maxval = orig.min(), orig.max()
print(minval, maxval)
```

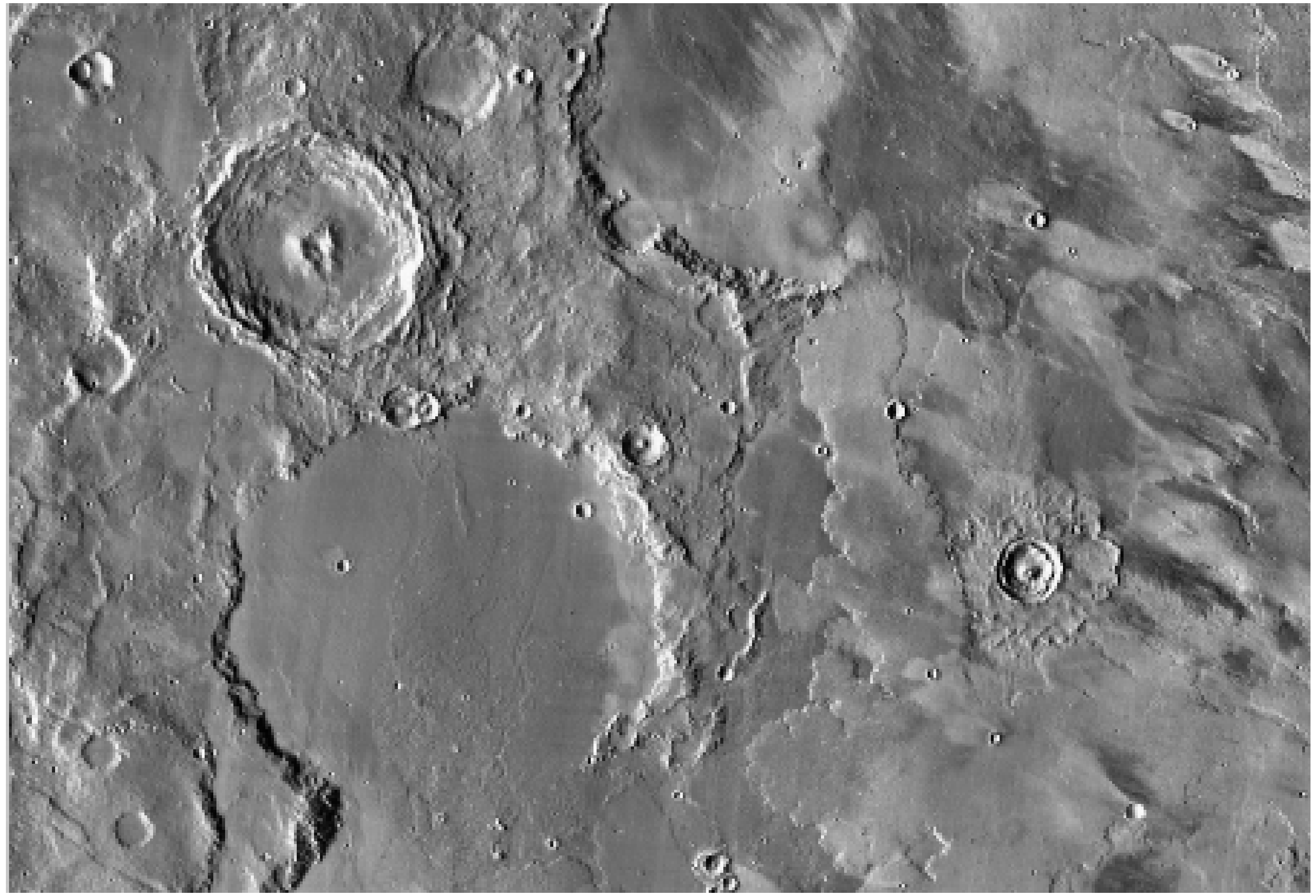
```
125 244
```

```
rescaled = (255/(maxval-minval)) * (pixels - minval)
print(rescaled.min(), rescaled.max())
```

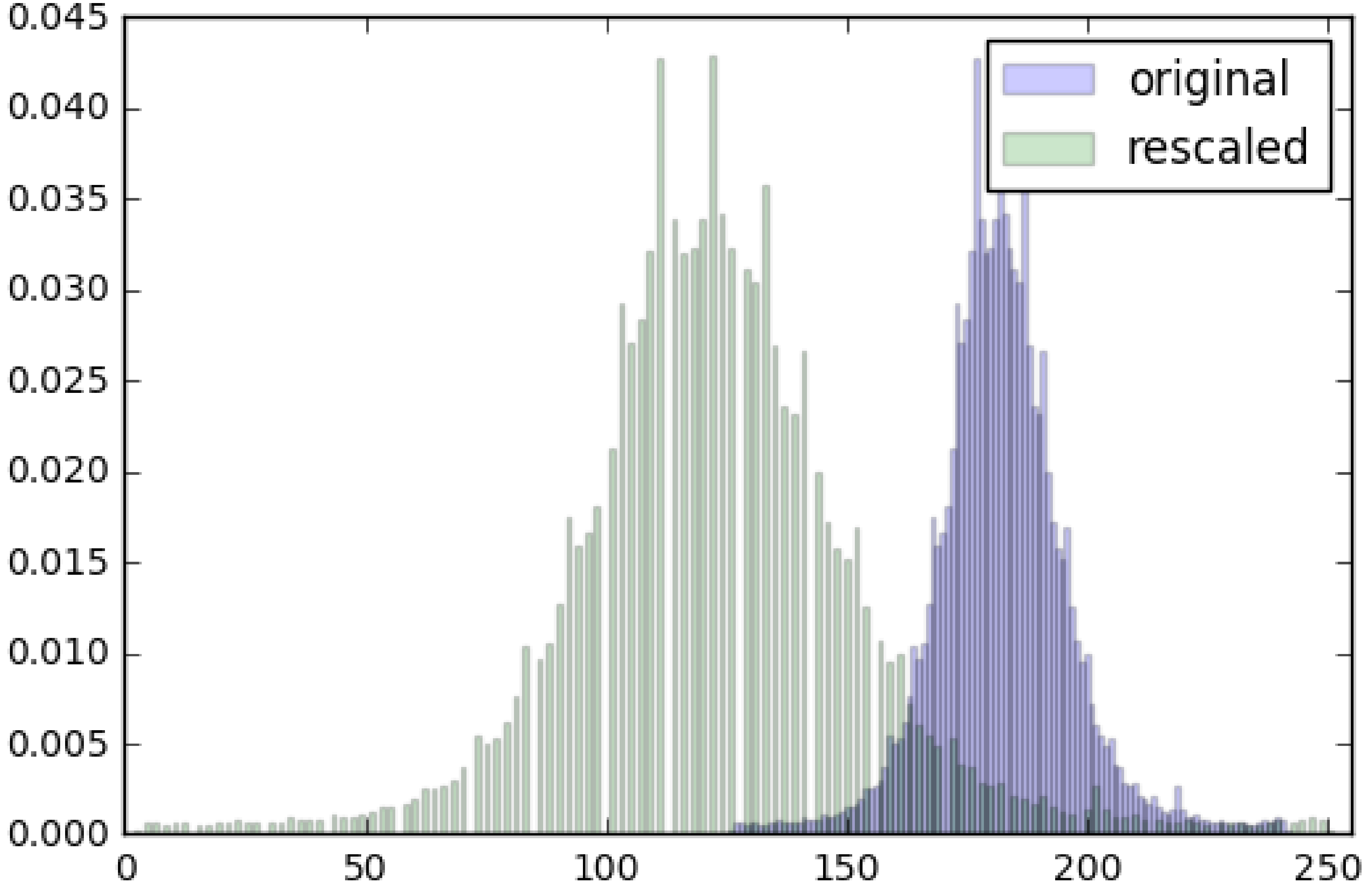
```
0.0 255.0
```

```
plt.imshow(rescaled)
plt.axis('off')
plt.show()
```

Rescaled image



Original and rescaled histograms



Original and rescaled histograms

```
plt.hist(orig.flatten(), bins=256,
             range=(0,255), normed=True,
             color='blue', alpha=0.2))
plt.hist(rescaled.flatten(), bins=256,
             range=(0,255), normed=True,
             color='green', alpha=0.2))
plt.legend(['original', 'rescaled'])
plt.show()
```

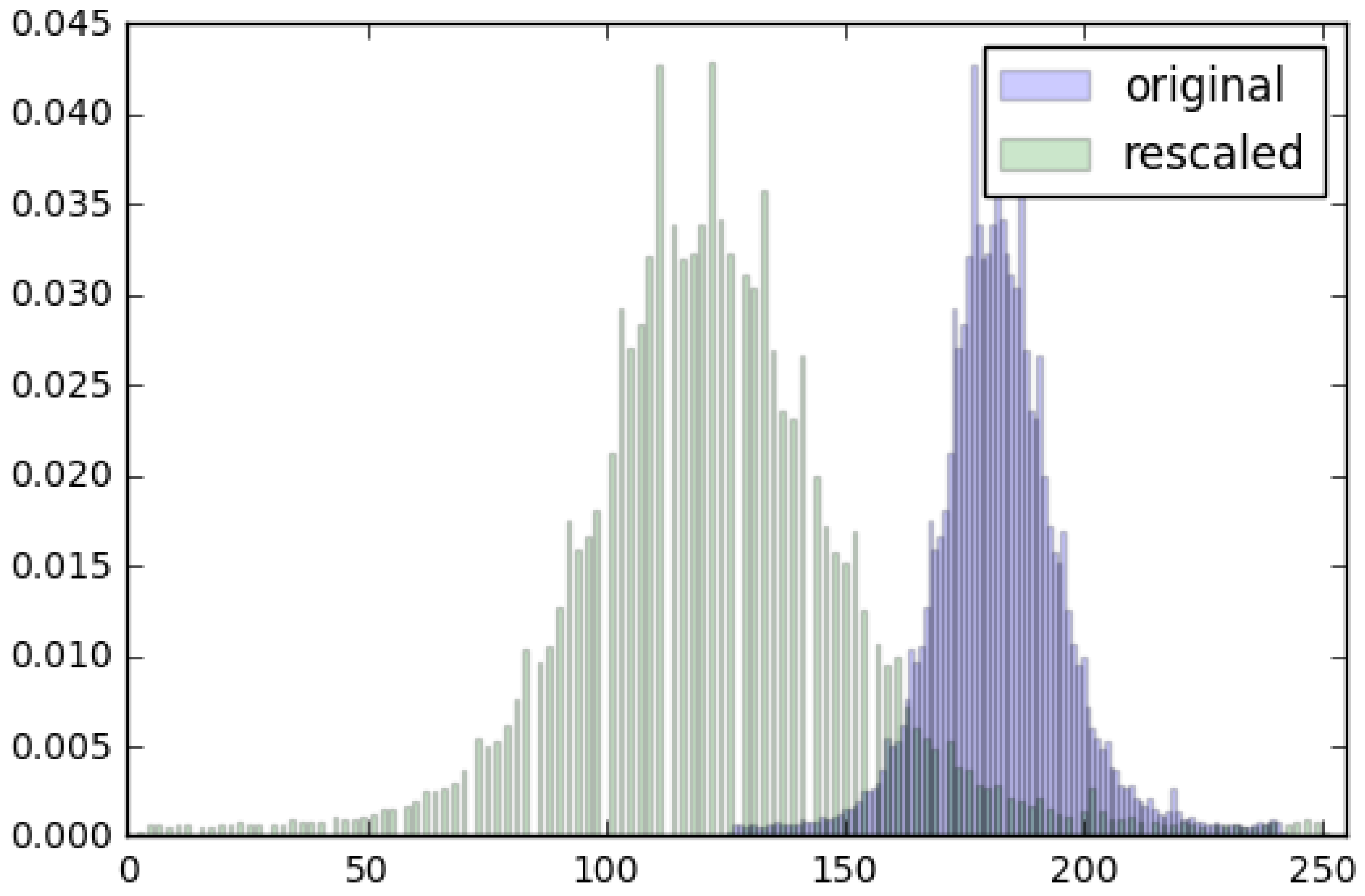


Image histogram & CDF

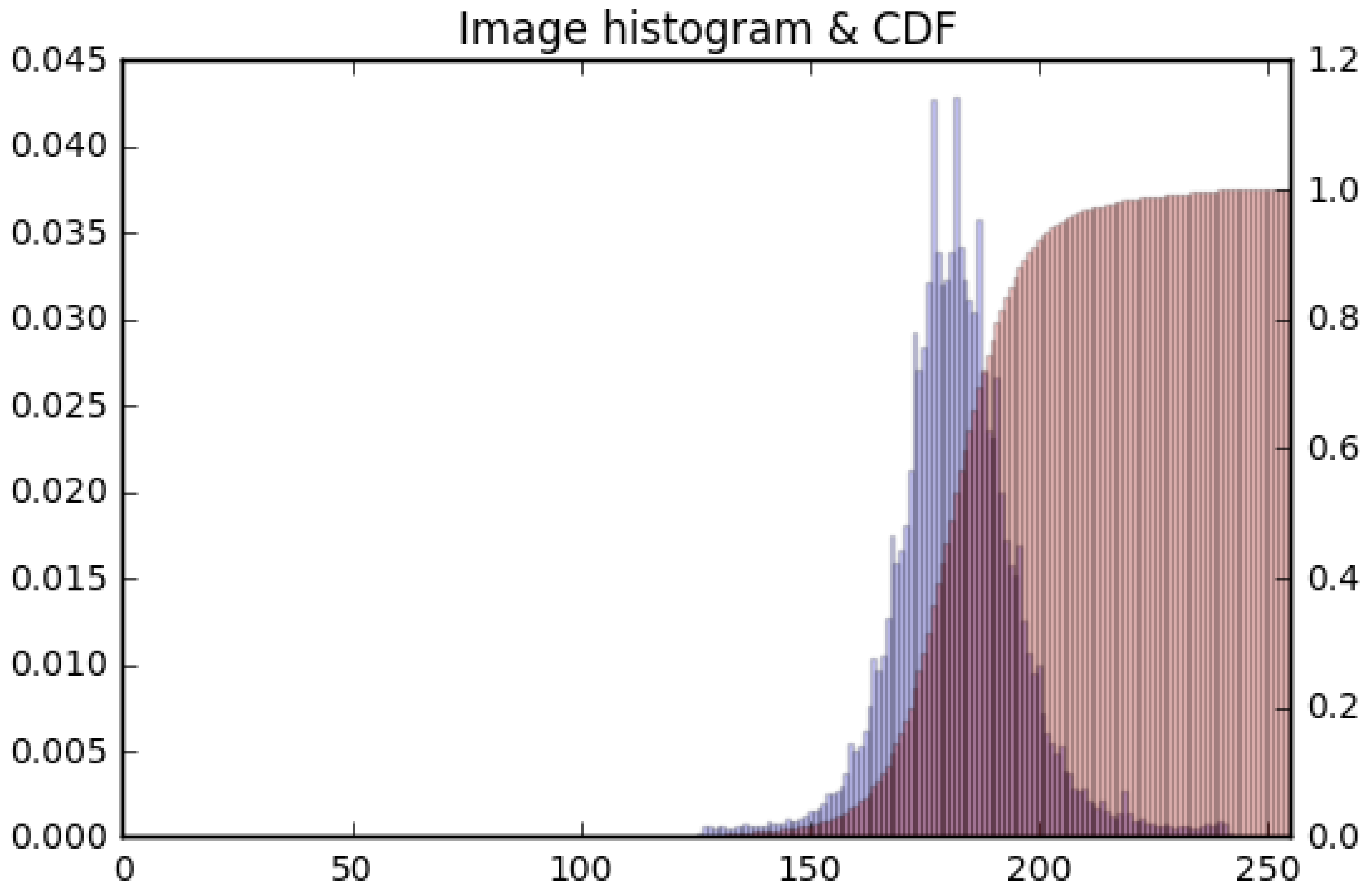
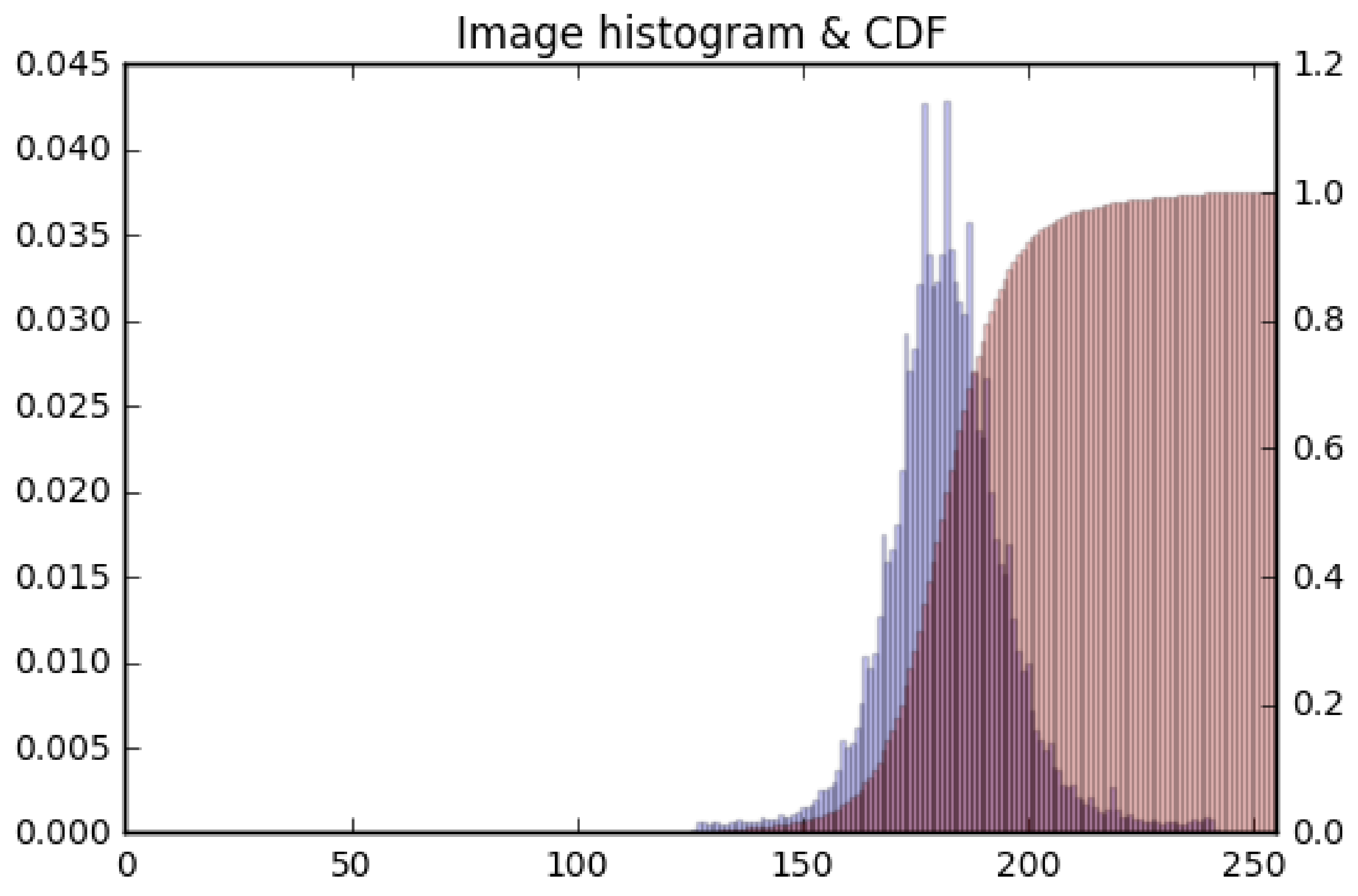


Image histogram & CDF

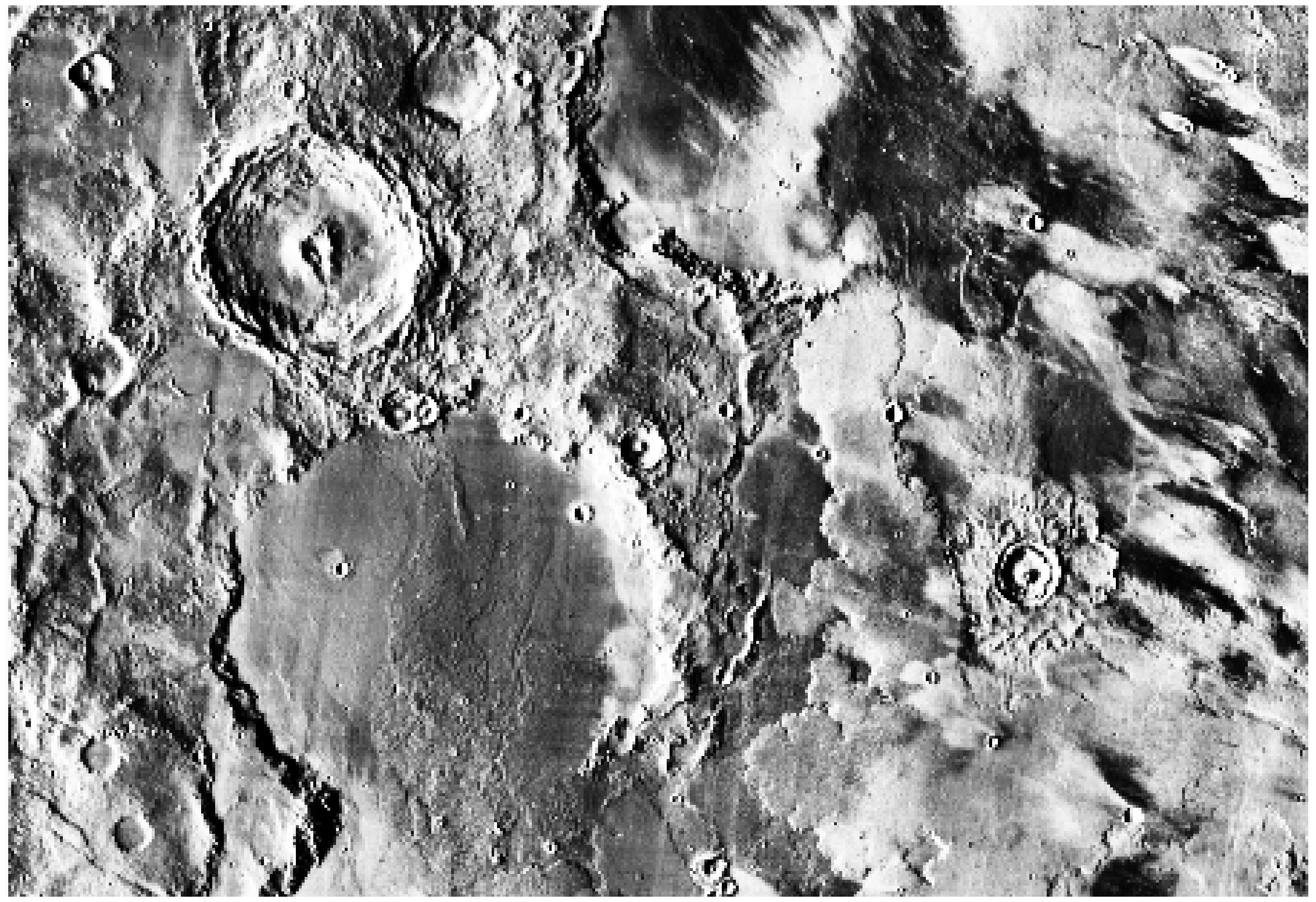
```
plt.hist(pixels, bins=256, range=(0,256),
               normed=True,
               color='blue', alpha=0.3)

plt.twinx()
orig_cdf, bins, patches = plt.hist(pixels,
    cumulative=True, bins=256,
    range=(0,256), normed=True,
    color='red', alpha=0.3)
plt.title('Image histogram and CDF')
plt.xlim((0, 255))
plt.show()
```



Equalizing intensity values

```
new_pixels =  np.interp(pixels,bins[:-1],
                              orig_cdf*255)
new = new_pixels.reshape(orig.shape)
plt.imshow(new)
plt.axis('off')
plt.title('Equalized image')
plt.show()
```

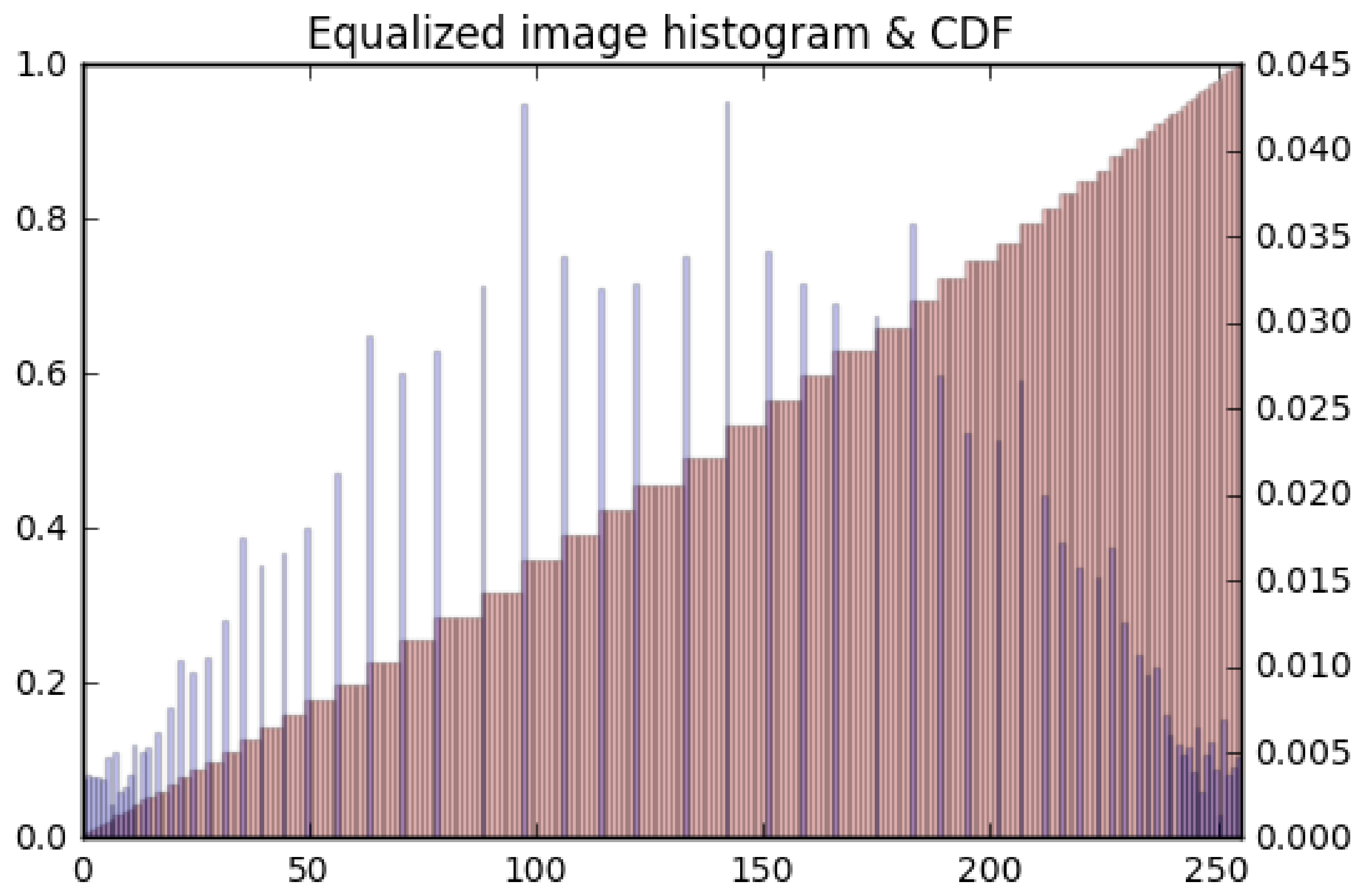
Equalized image



1,

Equalized histogram & CDF

```
plt.hist(new_pixels, bins=256, range=(0,256),
             normed=True,
             color='blue', alpha=0.3)
plt.twinx()
plt.hist(new_pixels, bins=256, range=(0,256),
             normed=True, cumulative=True,
             color='red', alpha=0.1)
plt.title('Equalized image histogram and CDF')
plt.xlim((0, 255))
plt.show()
```



Let's practice!

Congratulations!

INTRODUCTION TO DATA VISUALIZATION IN PYTHON

Bryan Van de Ven
Core Developer of Bokeh

Congratulations!

INTRODUCTION TO DATA VISUALIZATION IN PYTHON