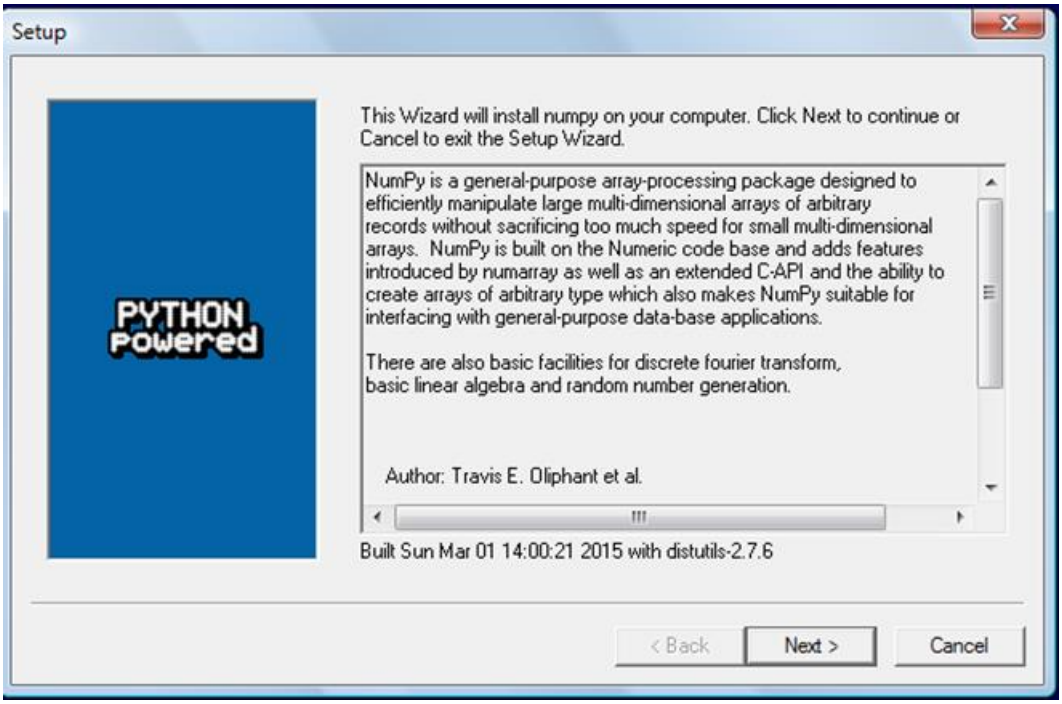
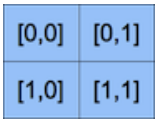
# 1

# NumPy Quick Start



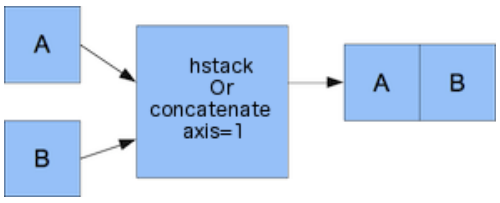

```
arange        arctan        argsort       array_equal   arrow
arccos        arctan2       argwhere      array_equiv
arccosh       arctanh       around        array_repr
arcsin        argmax        array         array_split
arcsinh       argmin        array2string  array_str
```

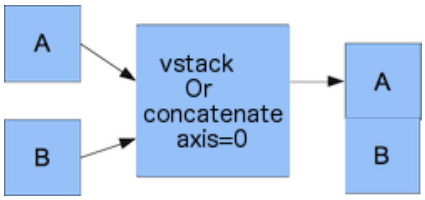
# 2
# Beginning with NumPy Fundamentals

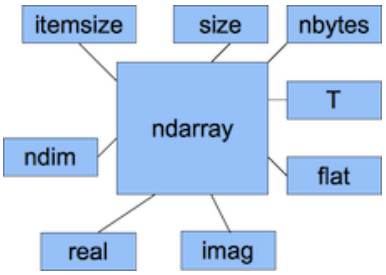



```
In [1]: int(42.0 + 1.j)
---------------------------------------
TypeError
<ipython-input-1-5e824780381a> in <modu
----> 1 int(42.0 + 1.j)

TypeError: can't convert complex to int
```




A
B
vstack
Or
concatenate
axis=0
A
B


itemsize
size
nbytes
ndarray
T
ndim
flat
real
imag

# 3

# Getting Familiar with Commonly Used Functions

$$\frac{1}{n}\sum_{i=1}^{n} a_i$$

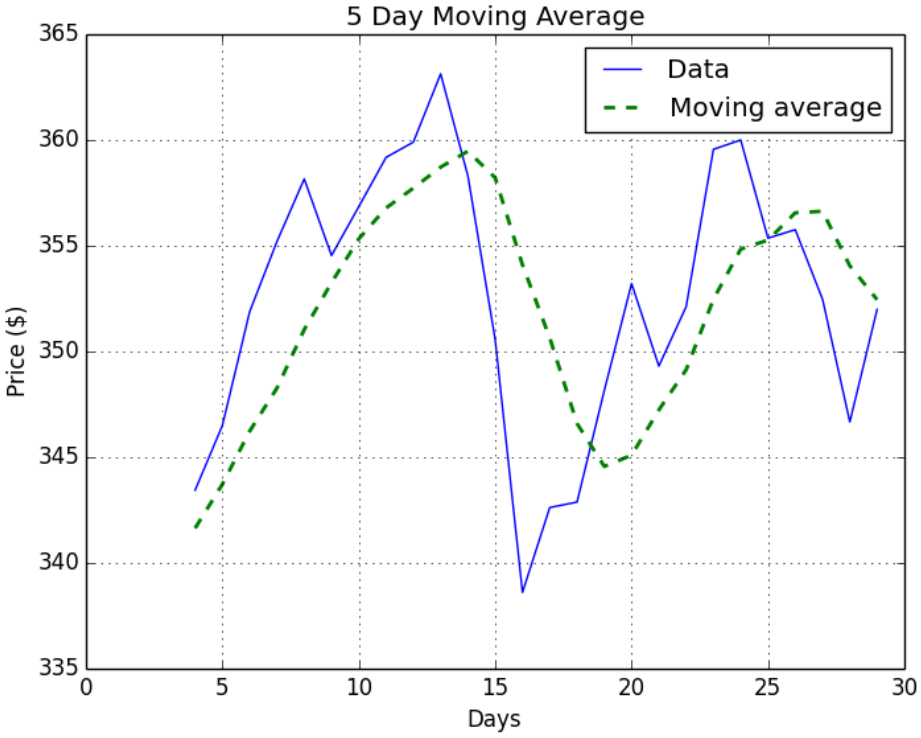
```
sorted = [ 336.1   338.61  339.32  342.62  342.88  343.44  344.32  345.03  346.5
  346.67  348.16  349.31  350.56  351.88  351.99  352.12  352.47  353.21
  354.54  355.2   355.36  355.76  356.85  358.16  358.3   359.18  359.56
  359.9   360.    363.13]
```

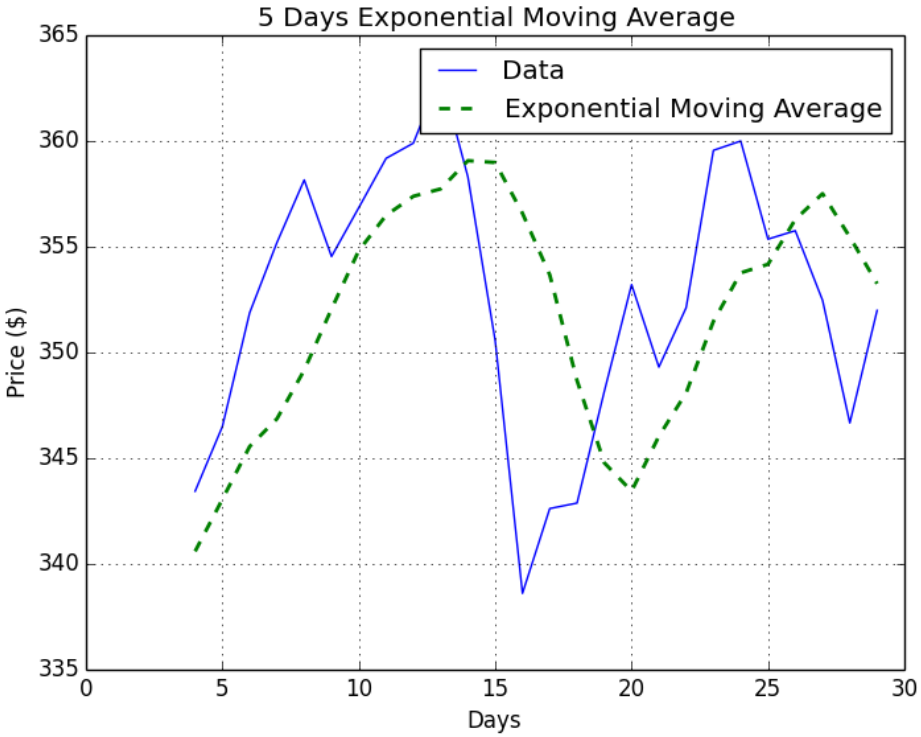
$$\frac{1}{n}\sum_{i=1}^{n}\left(a_i - mean\right)^2$$

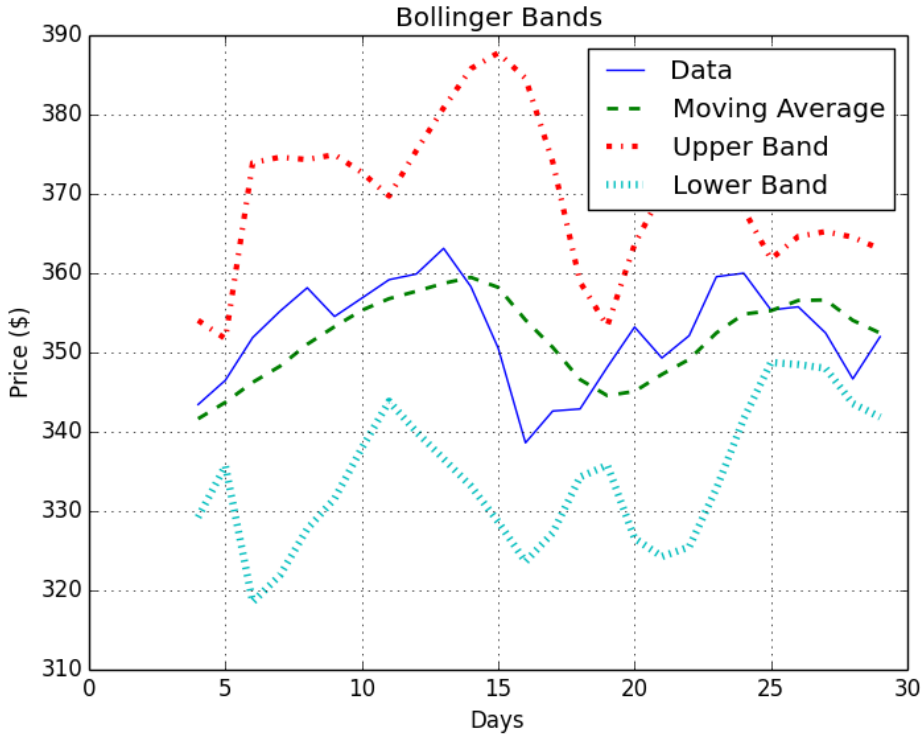
$$\log\left(a\right) - \log\left(b\right) = \log\left(\frac{a}{b}\right)$$

$$\frac{\left(\left(N-1\right)PATR + TR\right)}{N}$$

$$(f*g)(t)=\int_{-\infty}^{\infty} f(\tau)\,g(t-\tau)\,d\tau=\int_{-\infty}^{\infty} f(t-\tau)\,g(\tau)\,d\tau$$

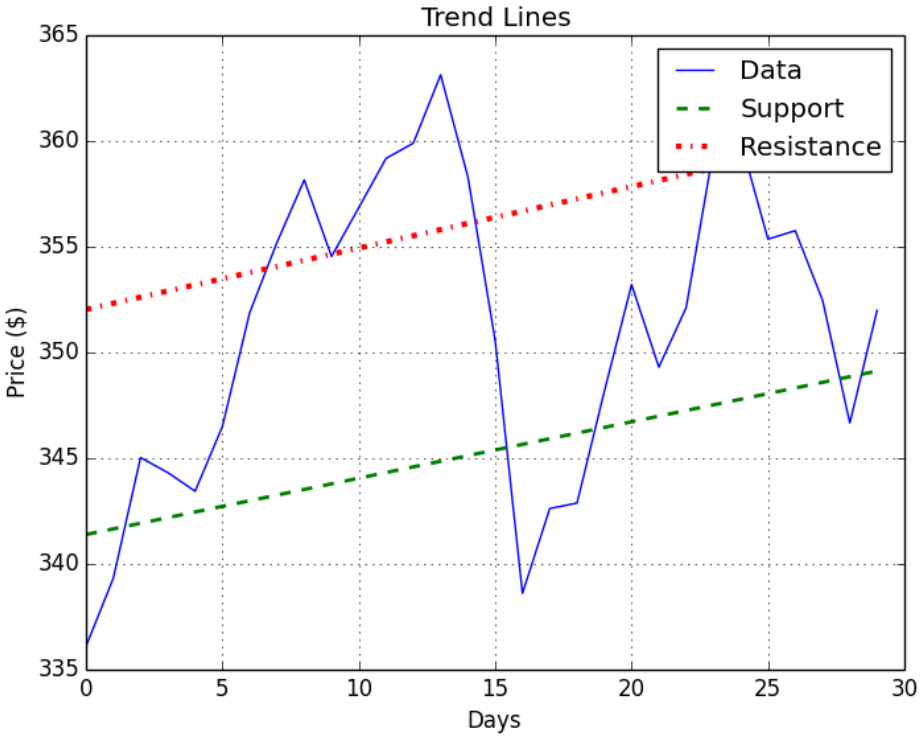



5 Days Exponential Moving Average
Data
Exponential Moving Average
Price ($)
Days
365
360
355
350
345
340
335
0
5
10
15
20
25
30




$$p_t = b + \sum_{i=1}^{N} a_{t-i} p_{t-i}$$

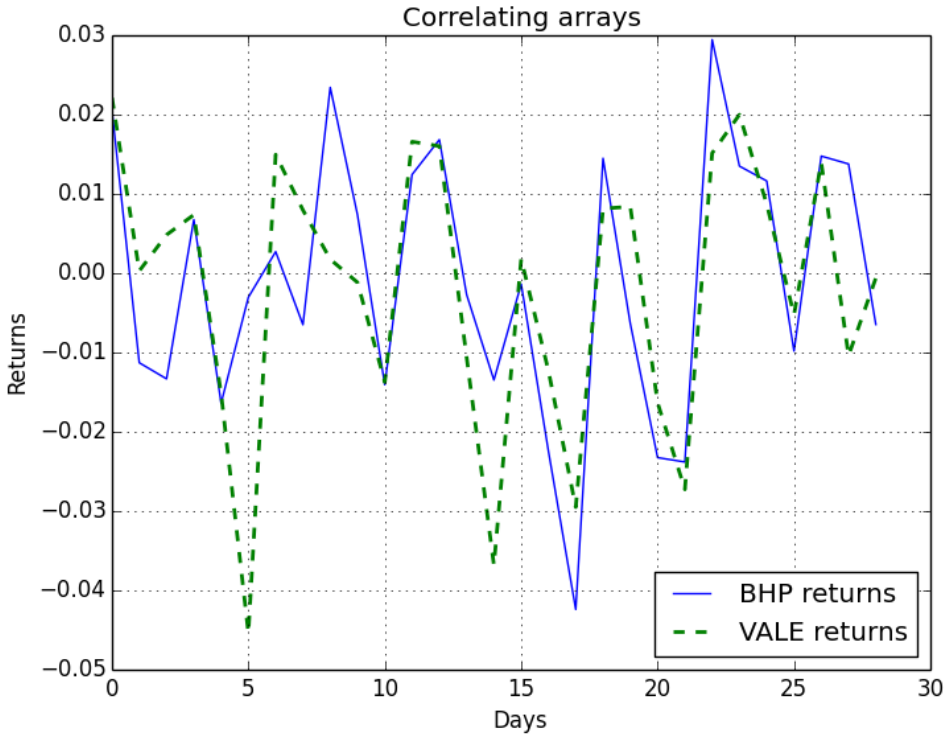
$$\sum_{i=1}^{n} \left( measured_i - predicted_i \right)^2$$


Trend Lines
Data
Support
Resistance
Price ($)
Days
365
360
355
350
345
340
335
0
5
10
15
20
25
30

# 4

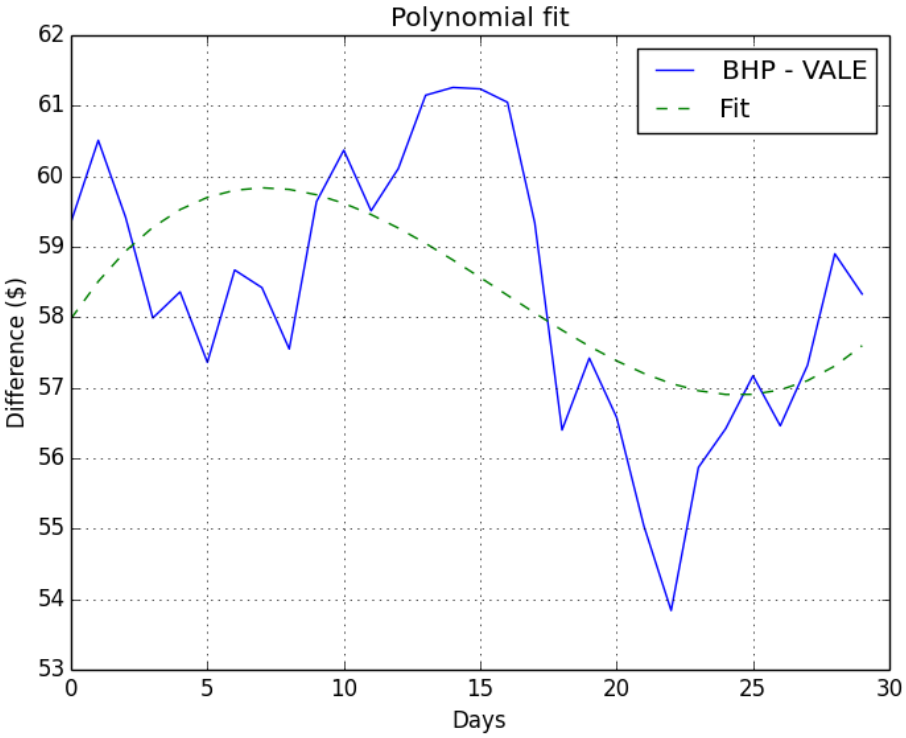
# Convenience Functions for your Convenience

$$cov(a,b) = \frac{1}{N}\sum_{i=1}^{N}\left(a_j - mean(a)\right)\left(b_j - mean(b)\right)$$

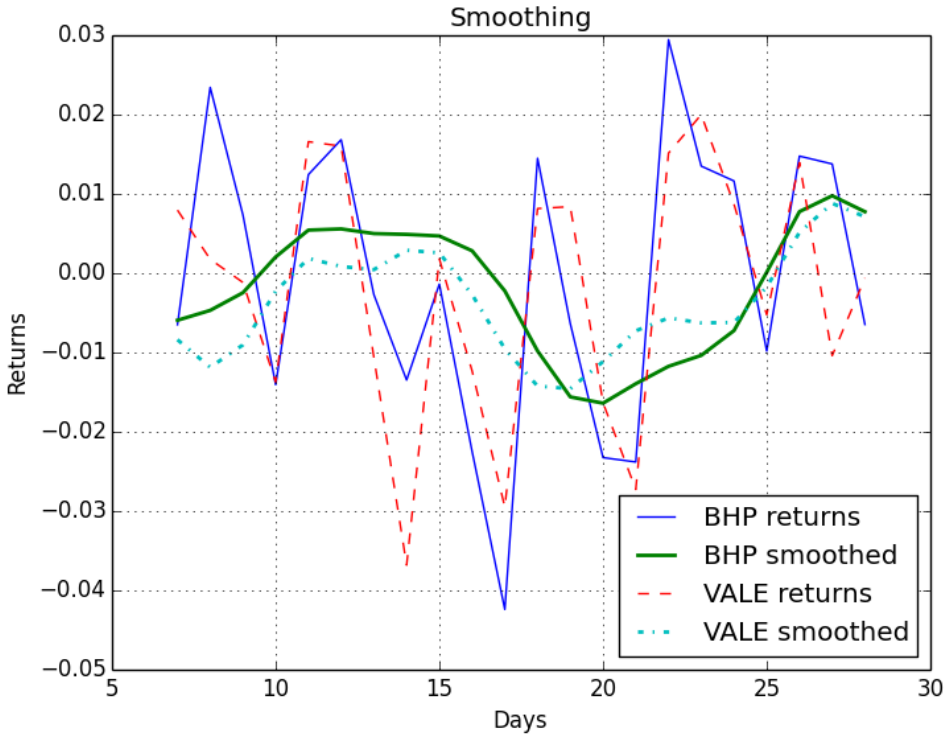



$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

$$f^{(n)}(a)$$




$$w(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right)$$

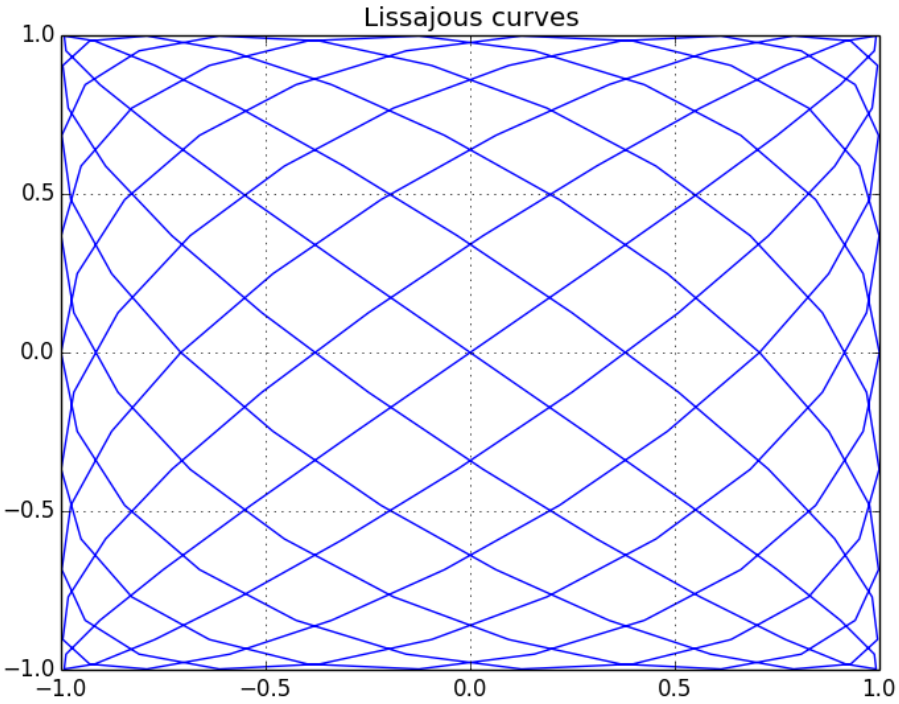

Smoothing
Returns
Days
0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
−0.04
−0.05
5
10
15
20
25
30
BHP returns
BHP smoothed
VALE returns
VALE smoothed

# 5
# Working with Matrices and ufuncs

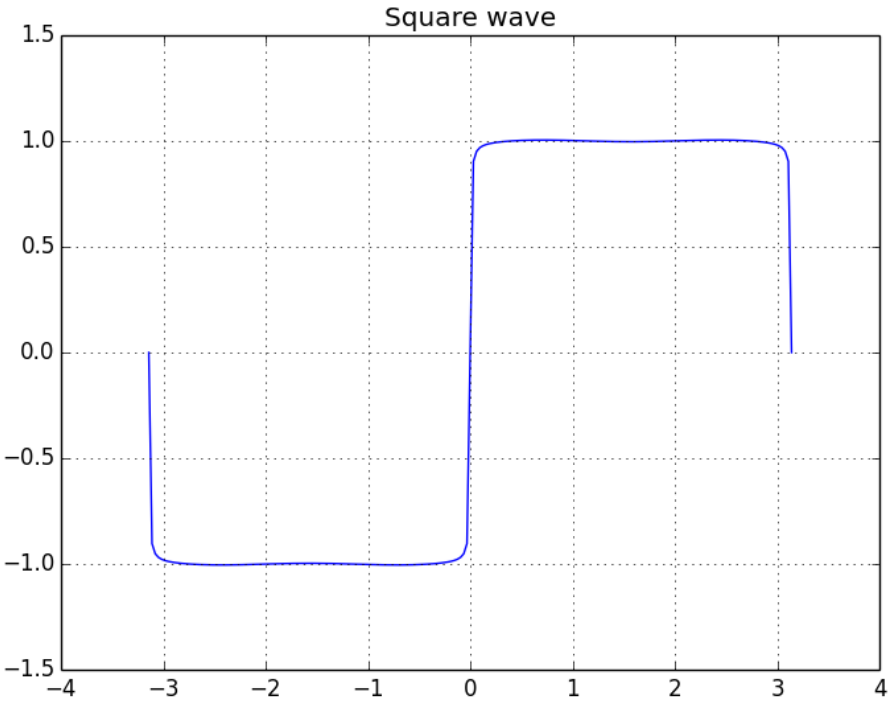
$$F_n = F_{n-1} + F_{n-2}$$

$$F_n = \frac{\varphi^n - (-\varphi)^{-n}}{\sqrt{5}}$$

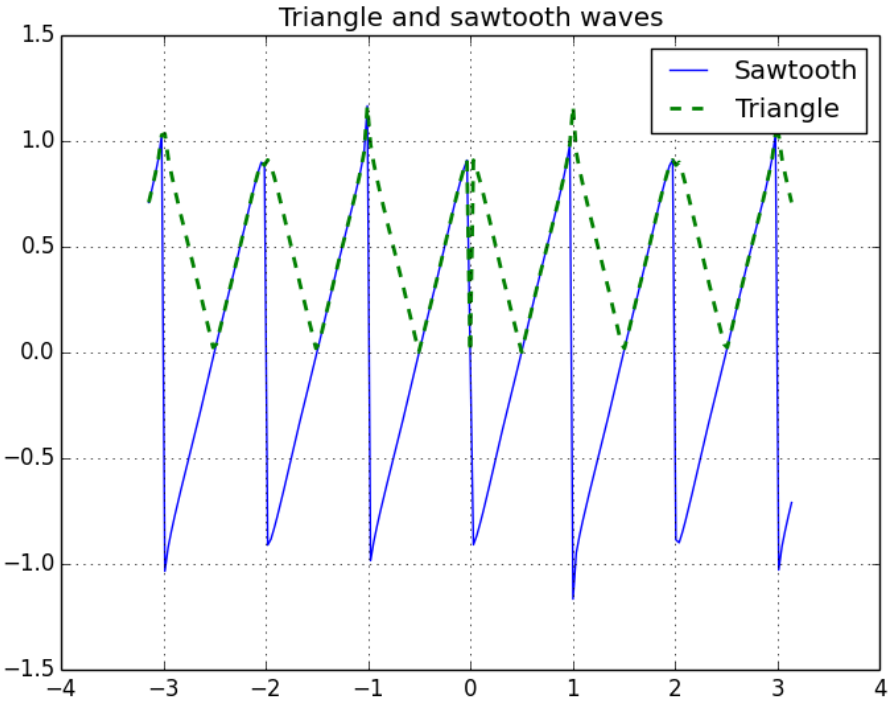
$$\varphi = \frac{1+\sqrt{5}}{2}$$




$$\sum_{k=1}^{\infty} \frac{4\sin\left(2\pi\left(2k-1\right)ft\right)}{\left(2k-1\right)\pi}$$




$$\sum_{k=1}^{\infty} \frac{-2\sin\left(2\pi kft\right)}{k\pi}$$

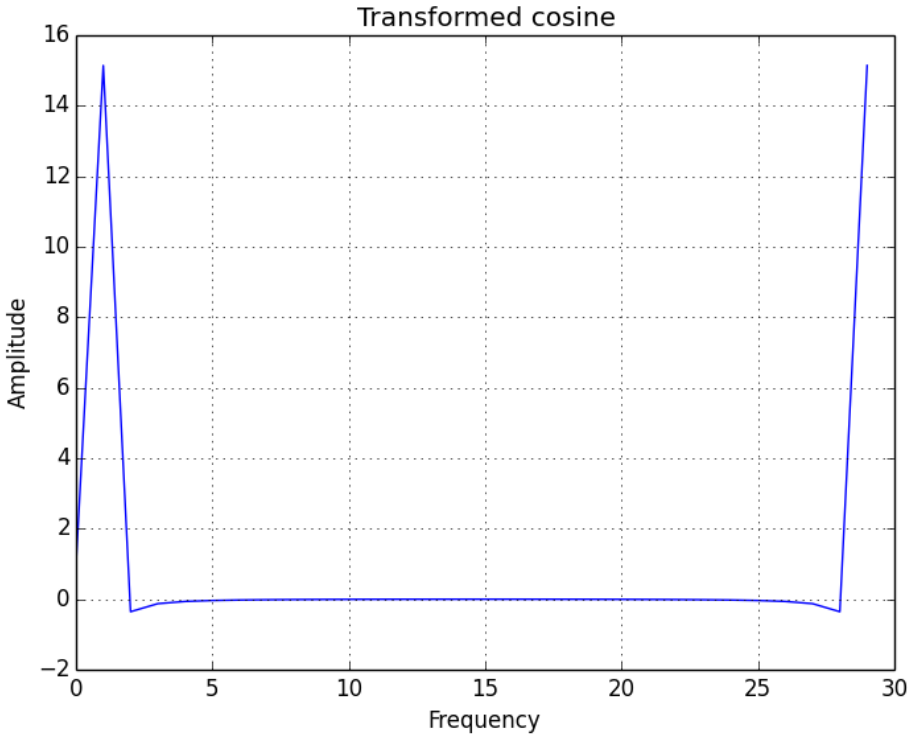

Triangle and sawtooth waves
Sawtooth
Triangle
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−4
−3
−2
−1
0
1
2
3
4

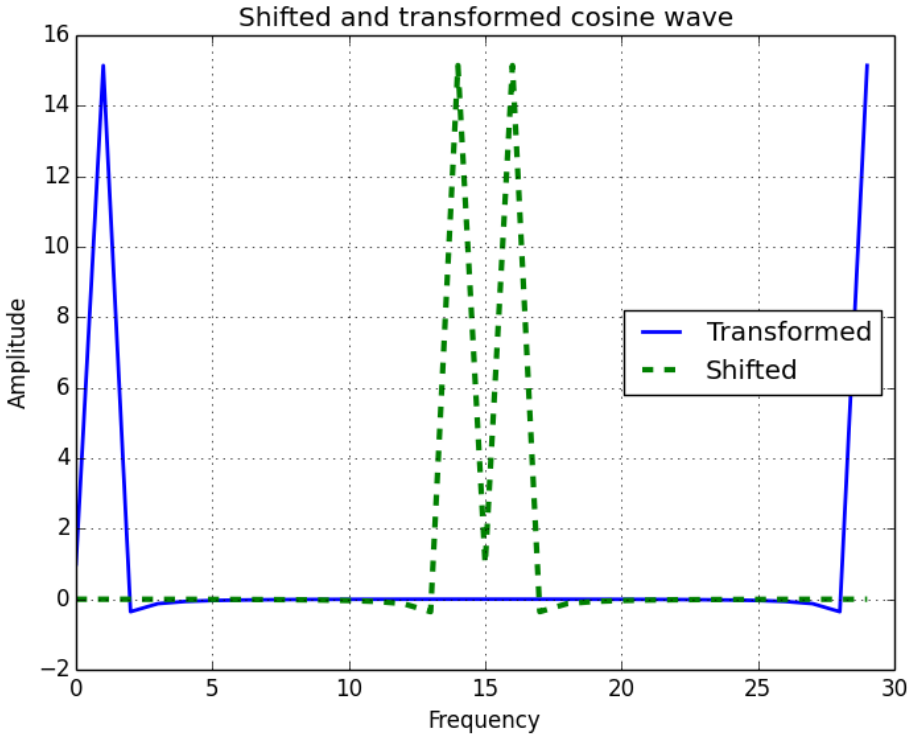
# 6
# Moving Further with NumPy Modules

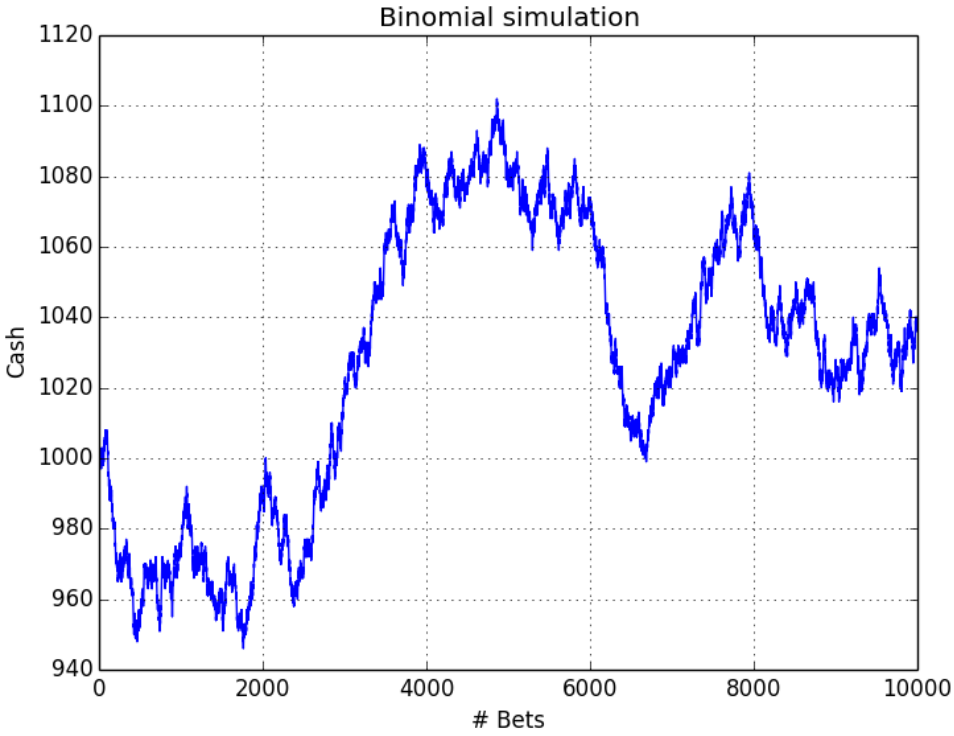
$$\sum_i A_{ij} B_i$$

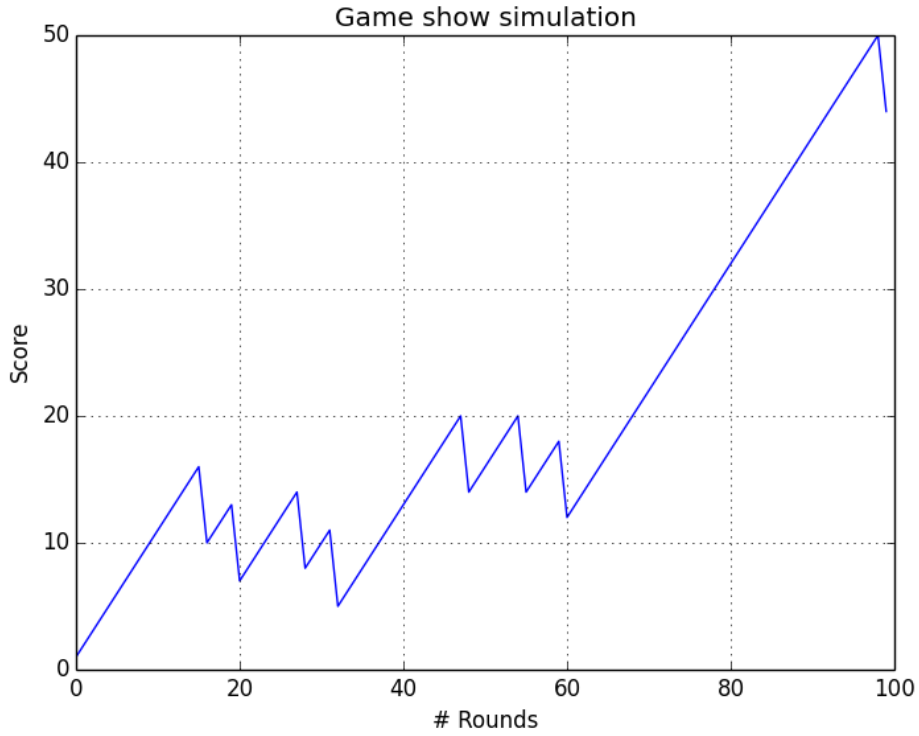
```
A
[[ 1 -2  1]
 [ 0  2 -8]
 [-4  5  9]]
b
[ 0  8 -9]
```

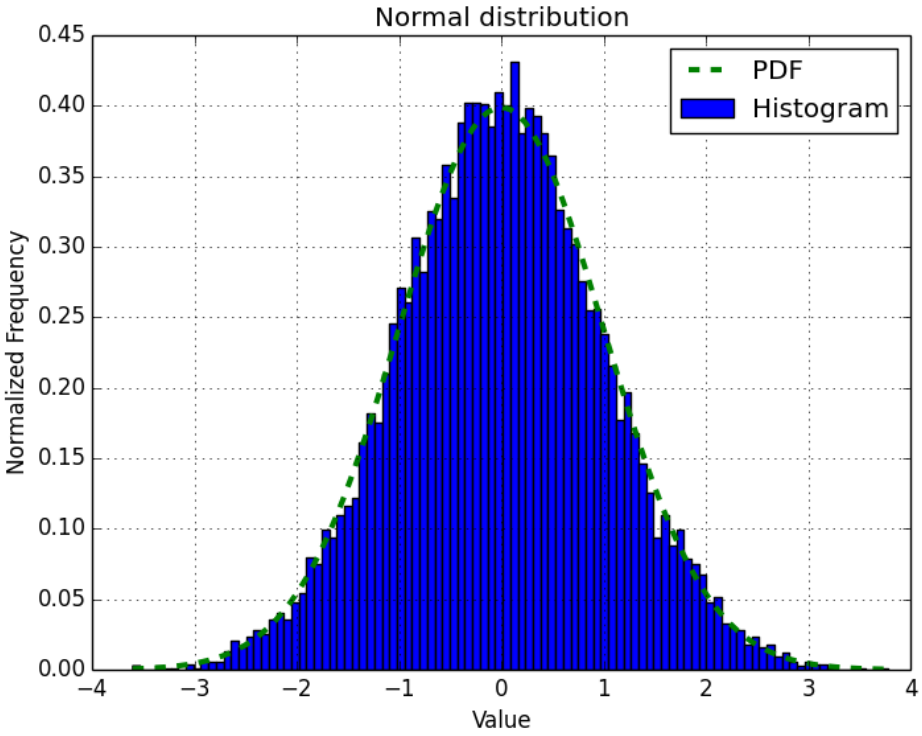
$$M = U \sum V^*$$

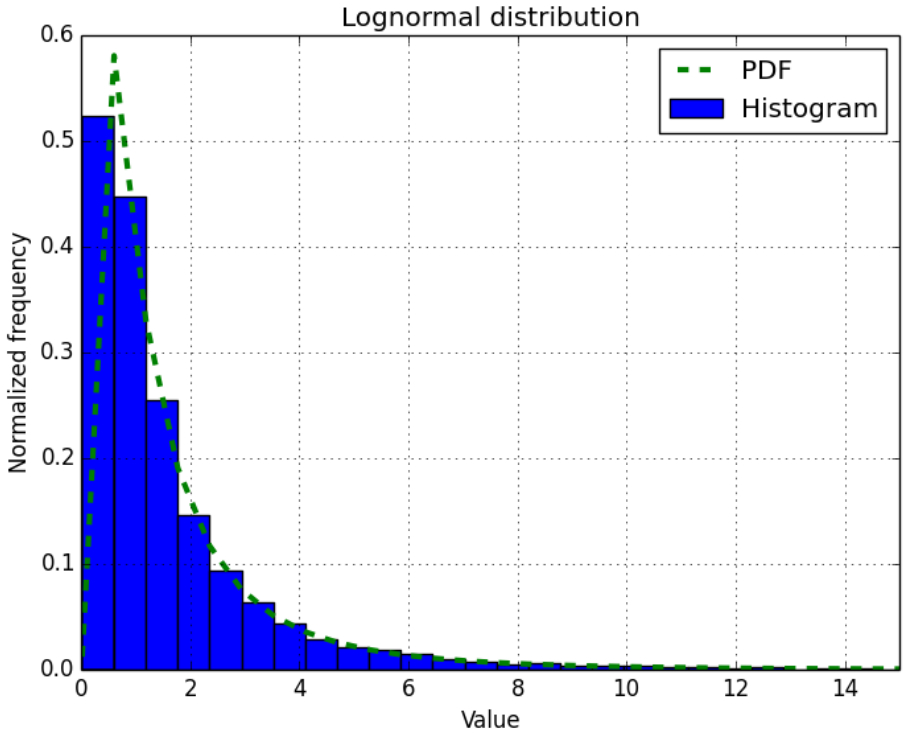

Transformed cosine
Amplitude
Frequency
16
14
12
10
8
6
4
2
0
−2
0
5
10
15
20
25
30


Shifted and transformed cosine wave
Amplitude
Frequency
Transformed
Shifted
16
14
12
10
8
6
4
2
0
−2
0
5
10
15
20
25
30


Binomial simulation
Cash
# Bets
1120
1100
1080
1060
1040
1020
1000
980
960
940
0
2000
4000
6000
8000
10000


Game show simulation
Score
# Rounds
0
10
20
30
40
50
0
20
40
60
80
100


Normal distribution
PDF
Histogram
Normalized Frequency
Value
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−3
−2
−1
0
1
2
3
4

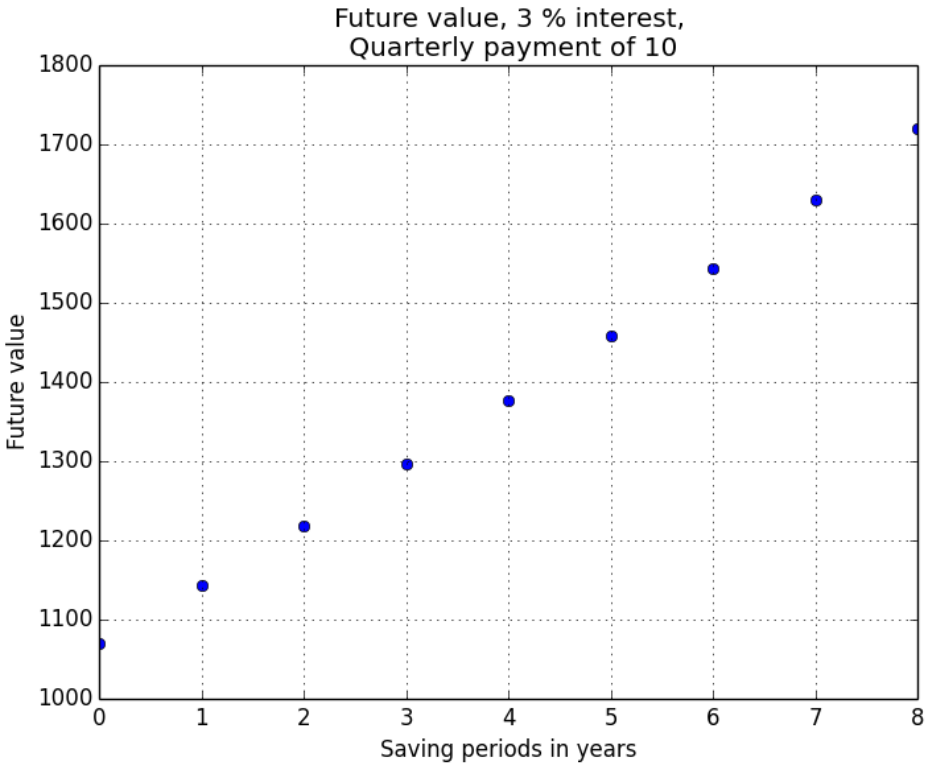



$$z = \frac{x - \mu}{\sigma}$$

# 7

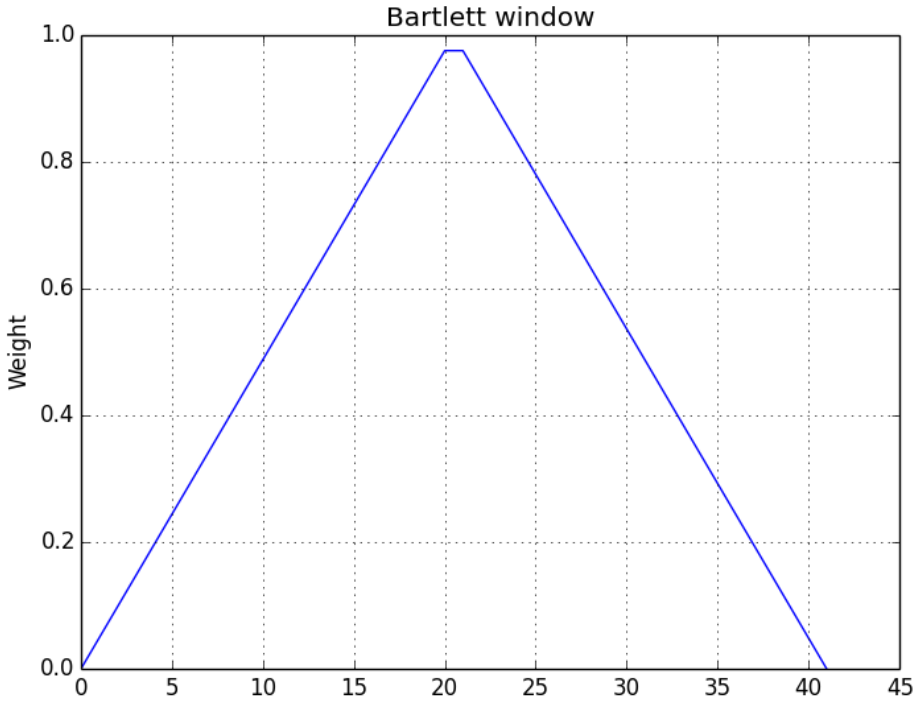
# Peeking Into Special Routines

$$PV\left(1+r\right)^{n}$$

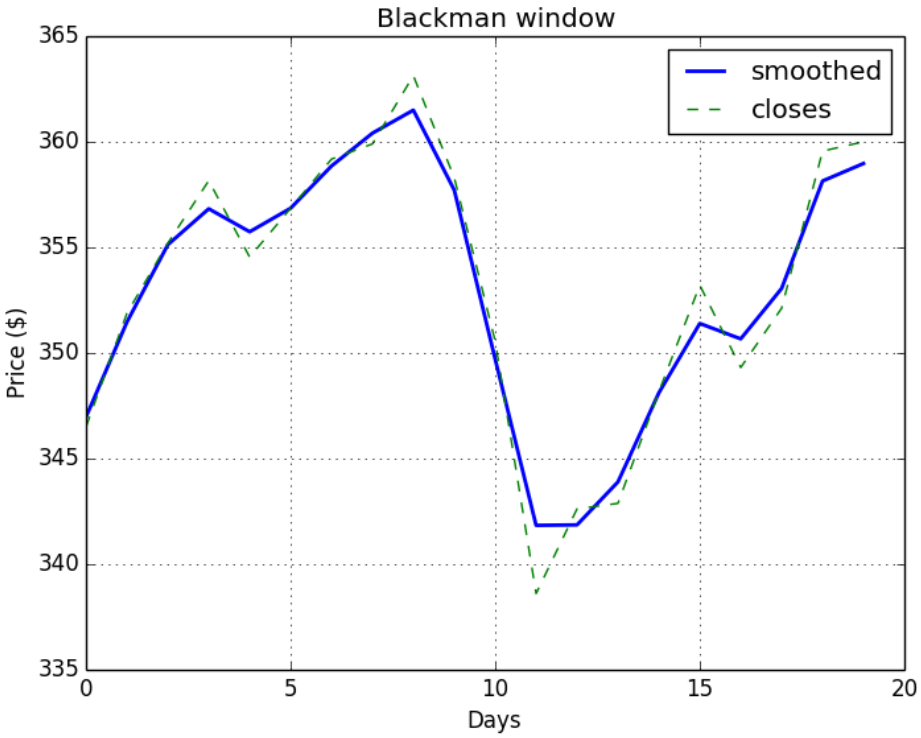


$$\sum_{t=0}^{N} \frac{R_t}{(1+r)^t}$$

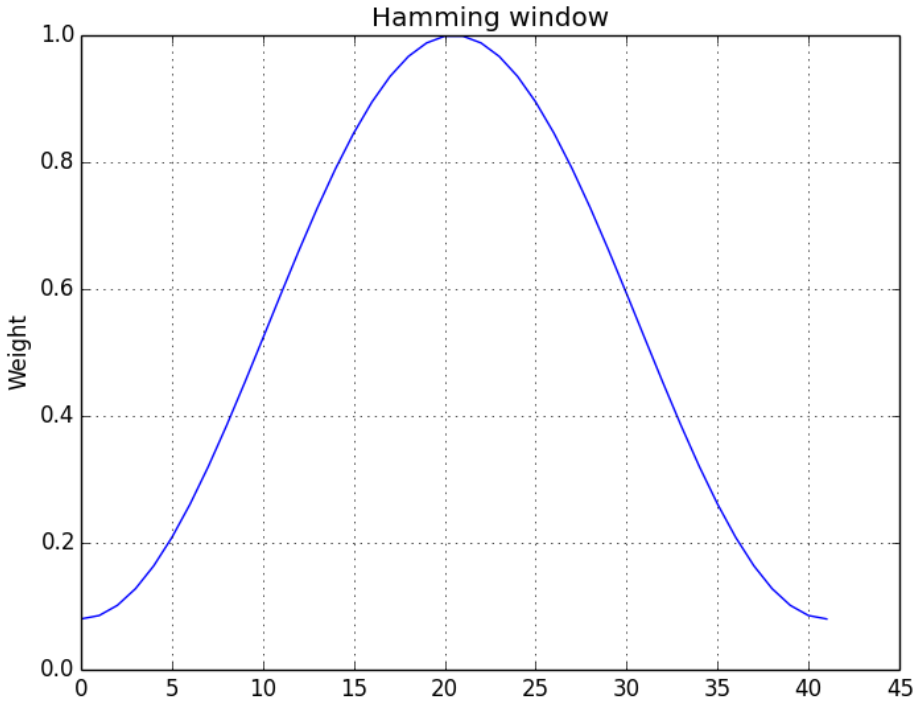
$$w(n) = 1 - \left| \frac{n - \frac{N-1}{2}}{\frac{N-1}{2}} \right|$$



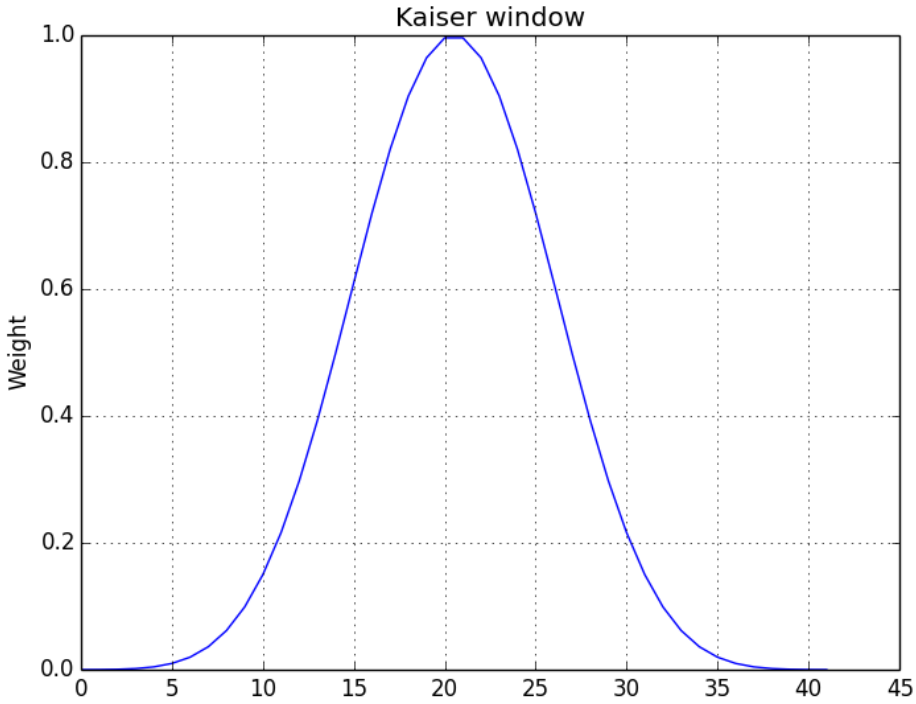
$$w(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{M}\right) + 0.08\cos\left(\frac{4\pi n}{M}\right)$$



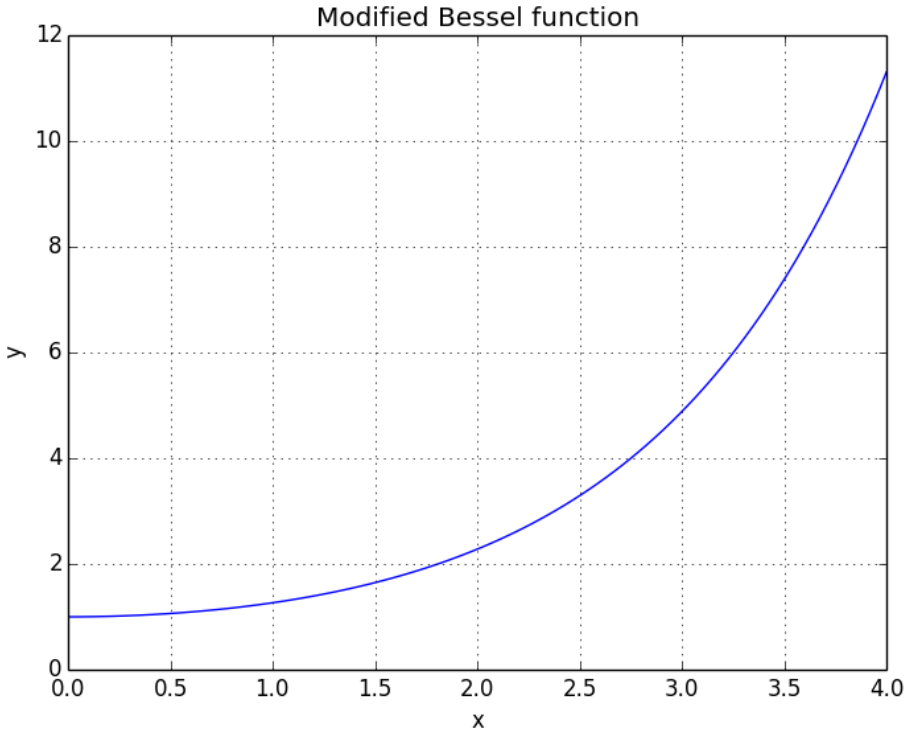

$$w(n) = 0.54 + 0.46\cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \le n \le M-1$$

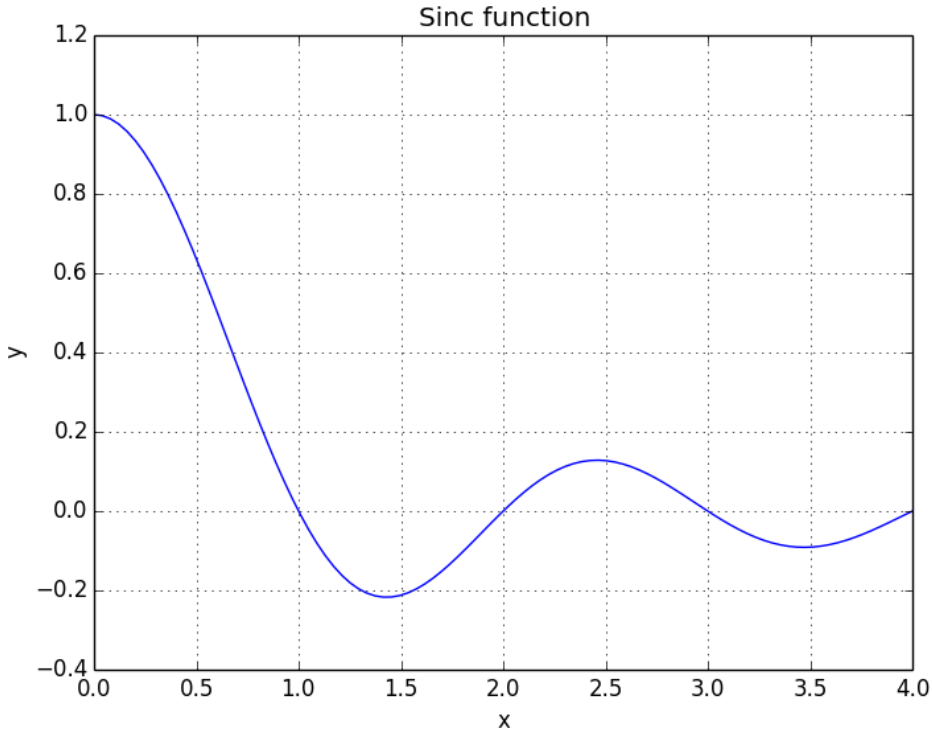


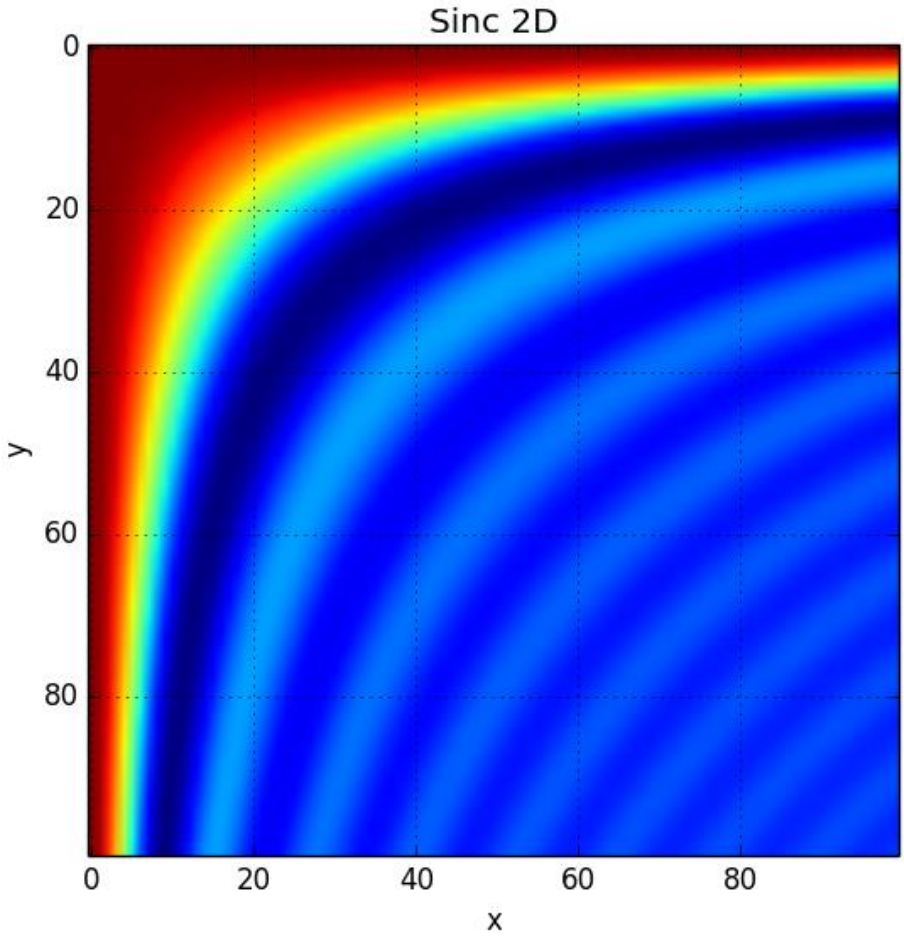

$$w(n) = I_0\left(\beta\sqrt{1-\frac{4n^2}{(M-1)^2}}\right) / I_0(\beta)$$




$$\frac{\sin(\pi x)}{\pi x}$$

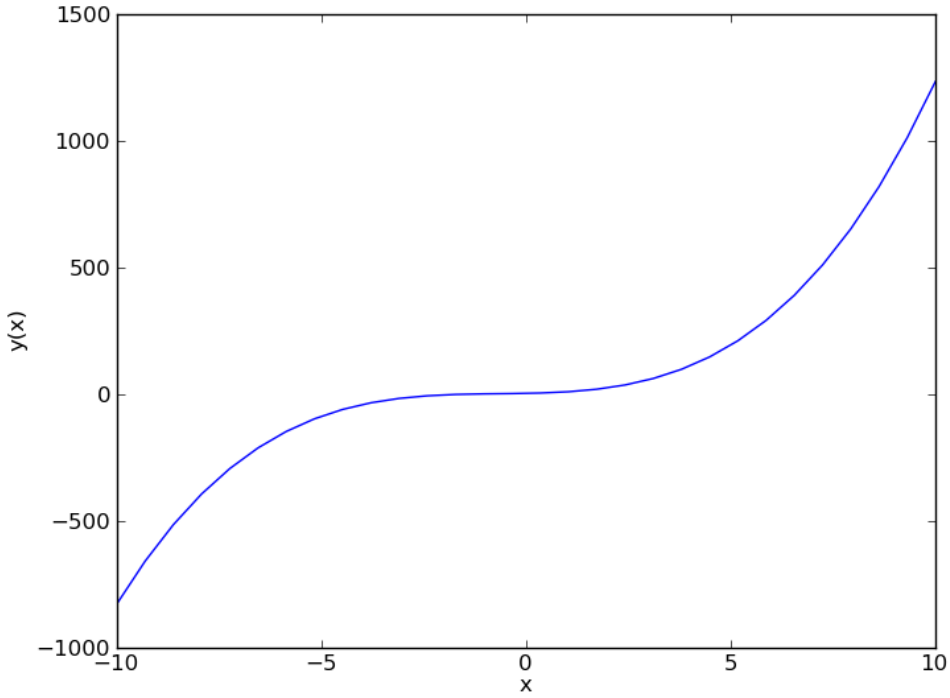

Modified Bessel function
12
10
8
6
4
2
0
y
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x

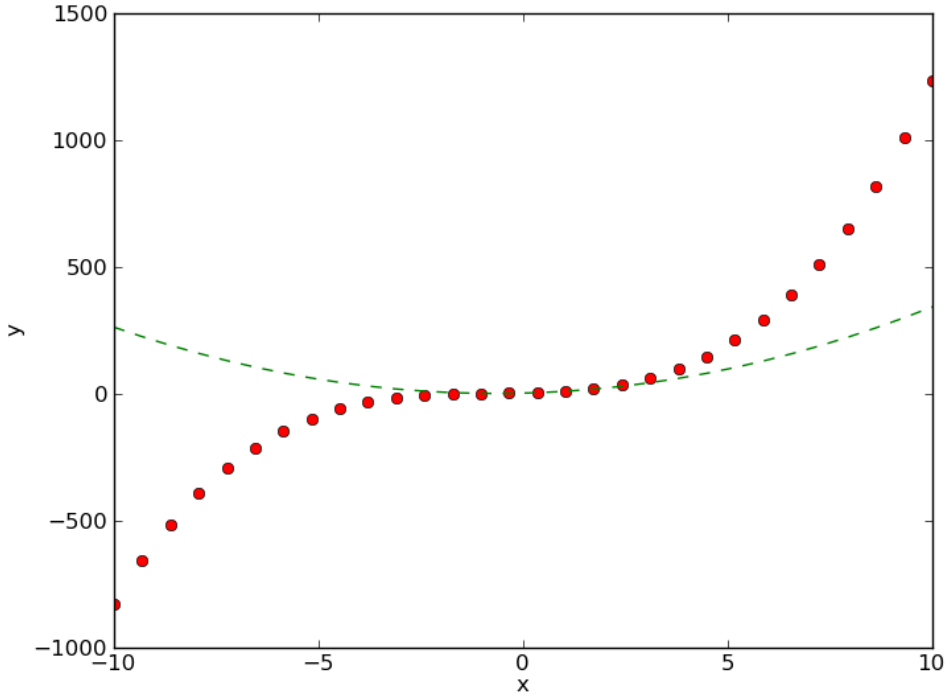

Sinc function
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
y
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x

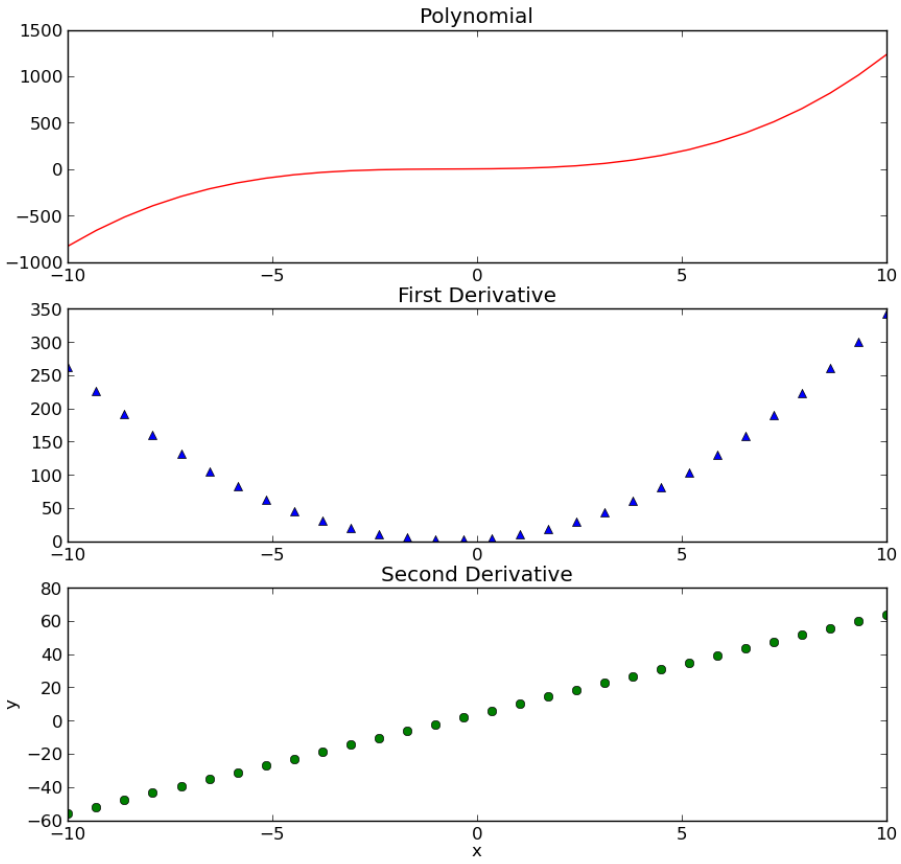

Sinc 2D
0
20
40
60
80
y
0
20
40
60
80
x

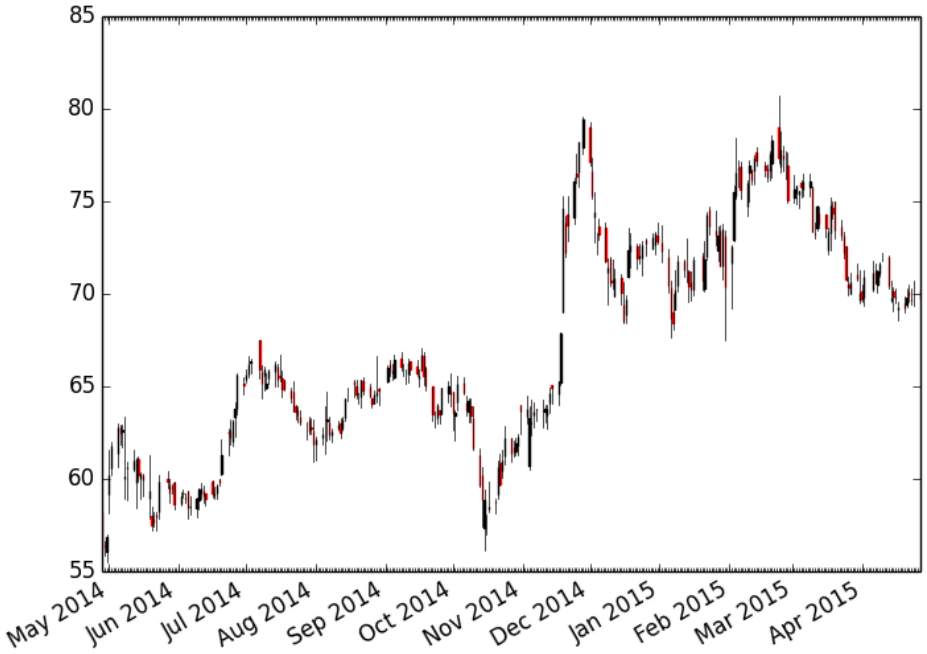
# 9

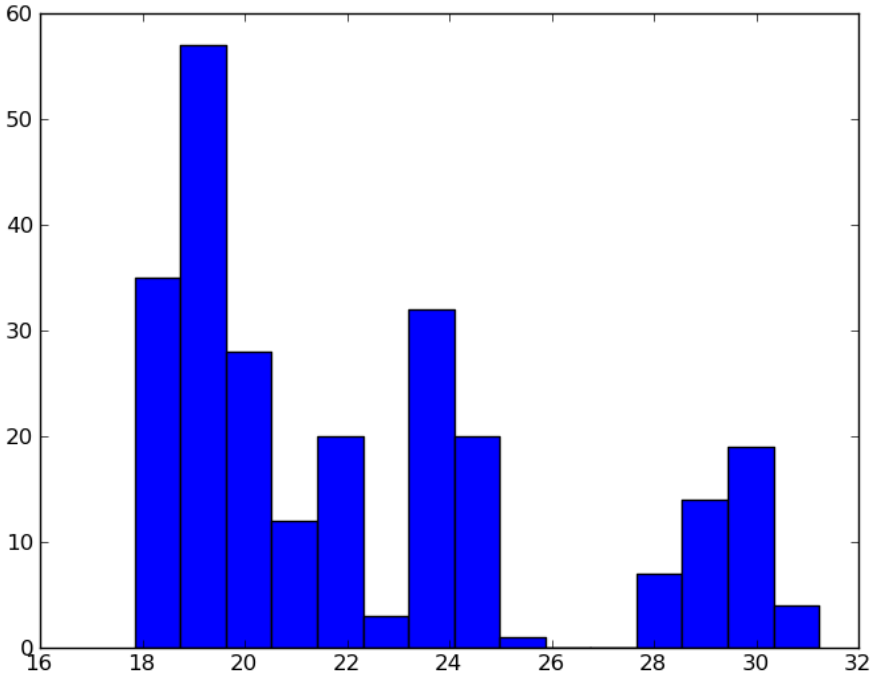
# Plotting with matplotlib

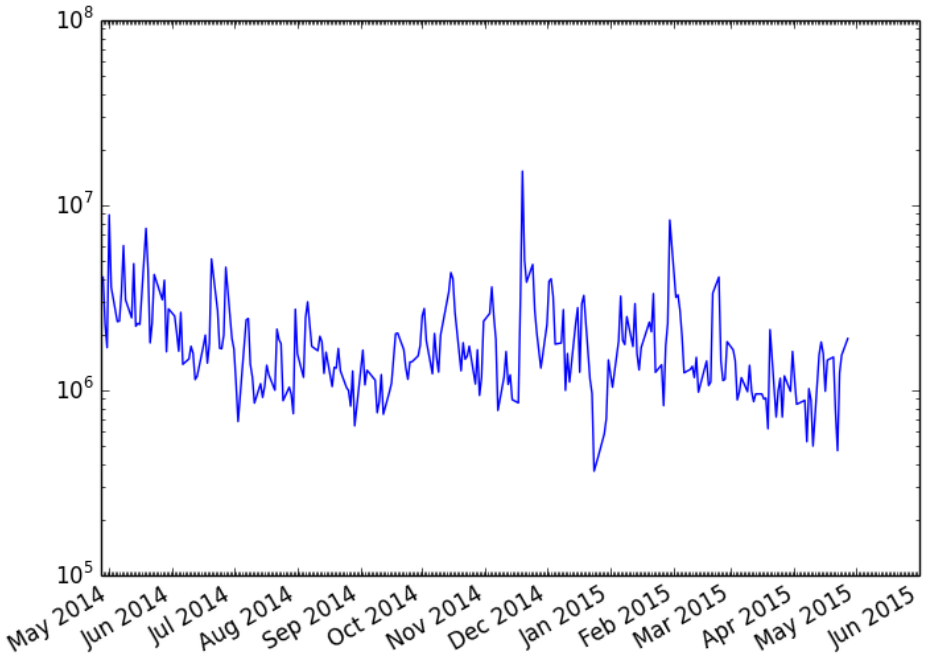


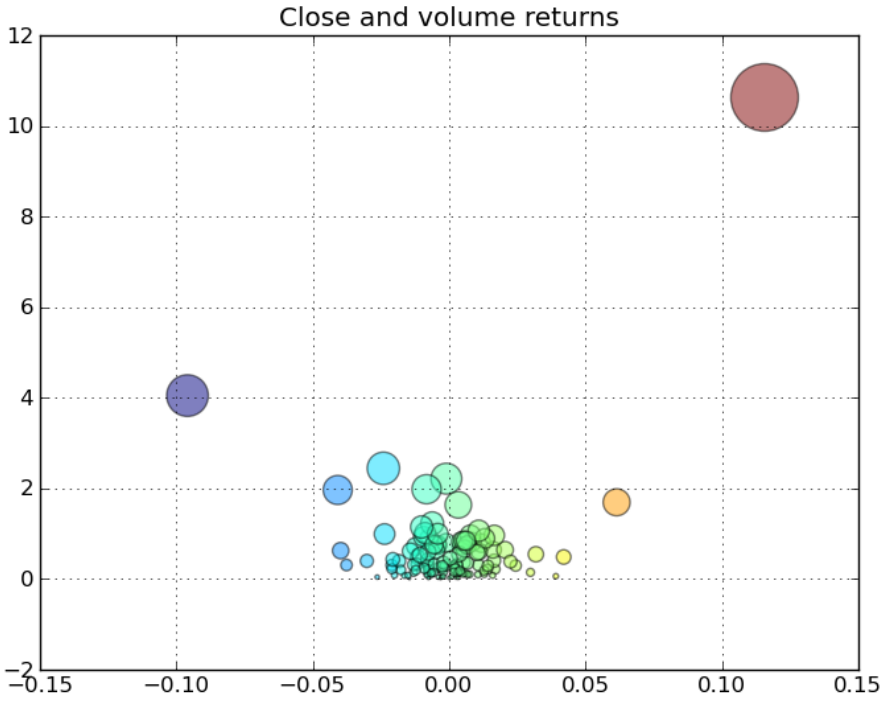

1500
1000
500
0
−500
−1000
y
−10
−5
0
5
10
x

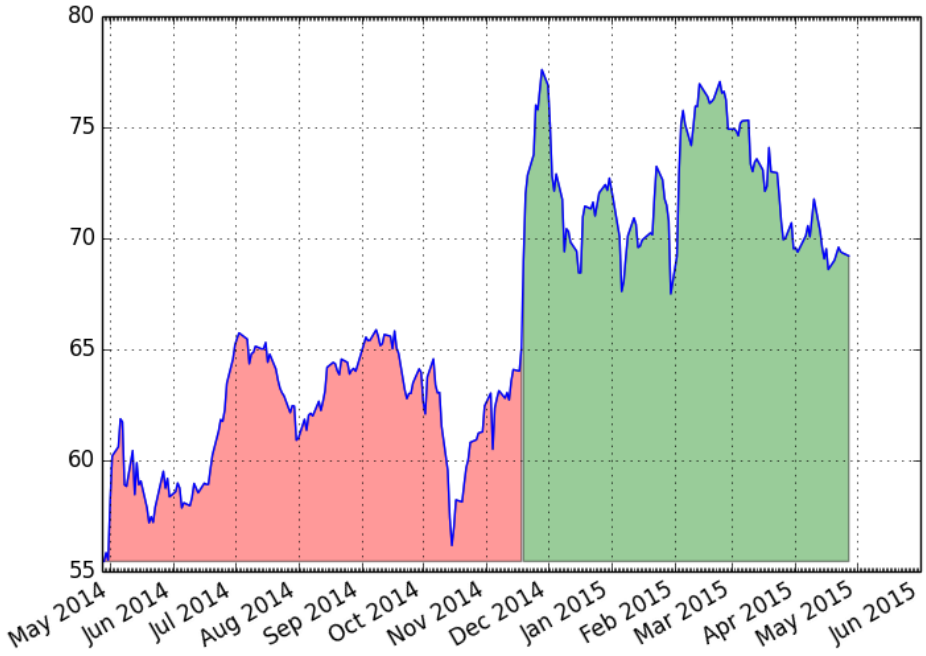

Polynomial
1500
1000
500
0
−500
−1000
−10
−5
0
5
10
First Derivative
350
300
250
200
150
100
50
0
−10
−5
0
5
10
Second Derivative
80
60
40
20
0
−20
−40
−60
y
−10
−5
0
5
10
x

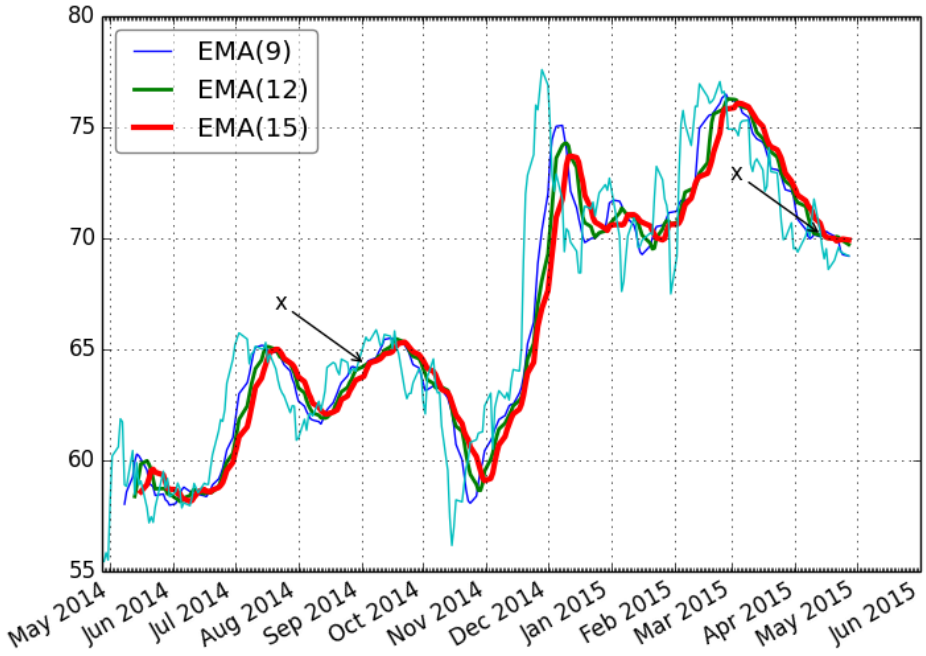

85
80
75
70
65
60
55
May 2014
Jun 2014
Jul 2014
Aug 2014
Sep 2014
Oct 2014
Nov 2014
Dec 2014
Jan 2015
Feb 2015
Mar 2015
Apr 2015


60
50
40
30
20
10
0
16
18
20
22
24
26
28
30
32

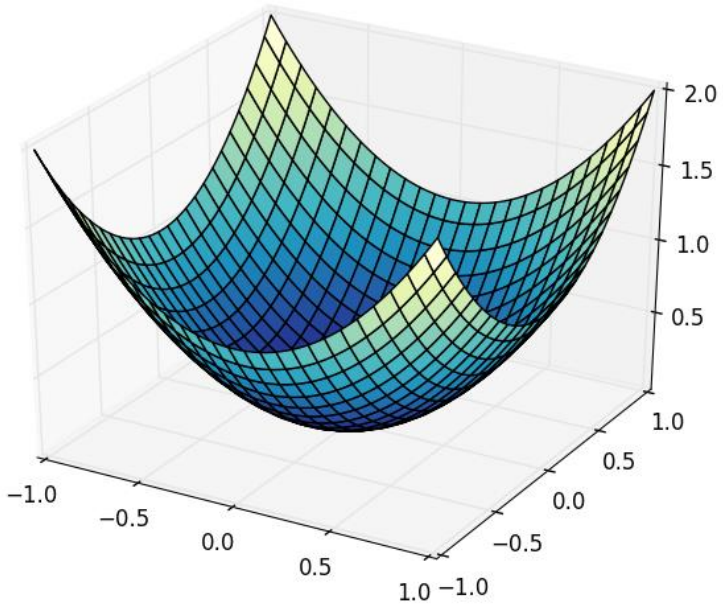

$10^8$
$10^7$
$10^6$
$10^5$
May 2014
Jun 2014
Jul 2014
Aug 2014
Sep 2014
Oct 2014
Nov 2014
Dec 2014
Jan 2015
Feb 2015
Mar 2015
Apr 2015
May 2015
Jun 2015

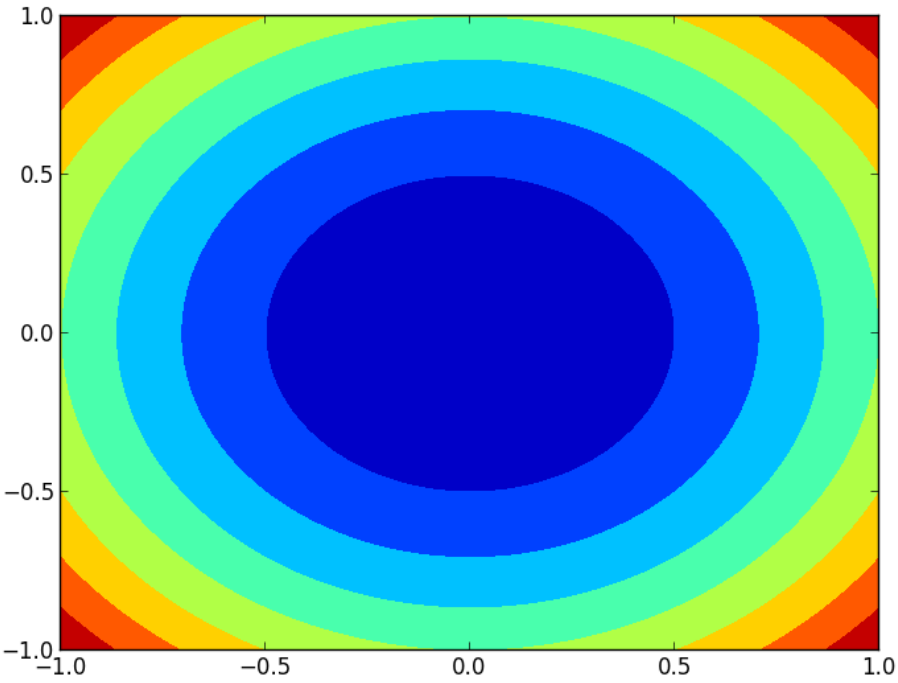

Close and volume returns
12
10
8
6
4
2
0
−2
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15

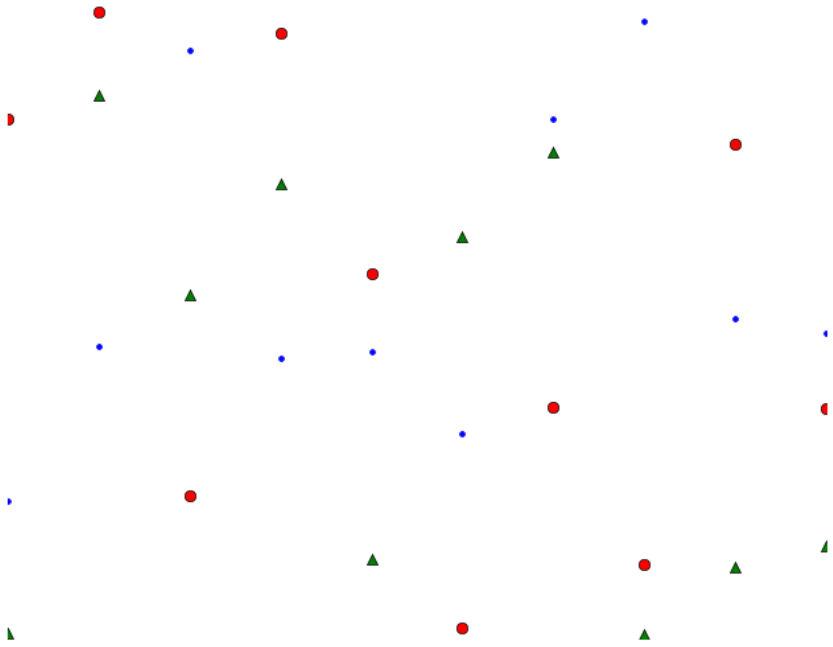

80
75
70
65
60
55
May 2014
Jun 2014
Jul 2014
Aug 2014
Sep 2014
Oct 2014
Nov 2014
Dec 2014
Jan 2015
Feb 2015
Mar 2015
Apr 2015
May 2015
Jun 2015


EMA(9)
EMA(12)
EMA(15)
80
75
70
65
60
55
x
x
May 2014
Jun 2014
Jul 2014
Aug 2014
Sep 2014
Oct 2014
Nov 2014
Dec 2014
Jan 2015
Feb 2015
Mar 2015
Apr 2015
May 2015
Jun 2015

$$z = x^2 + y^2$$

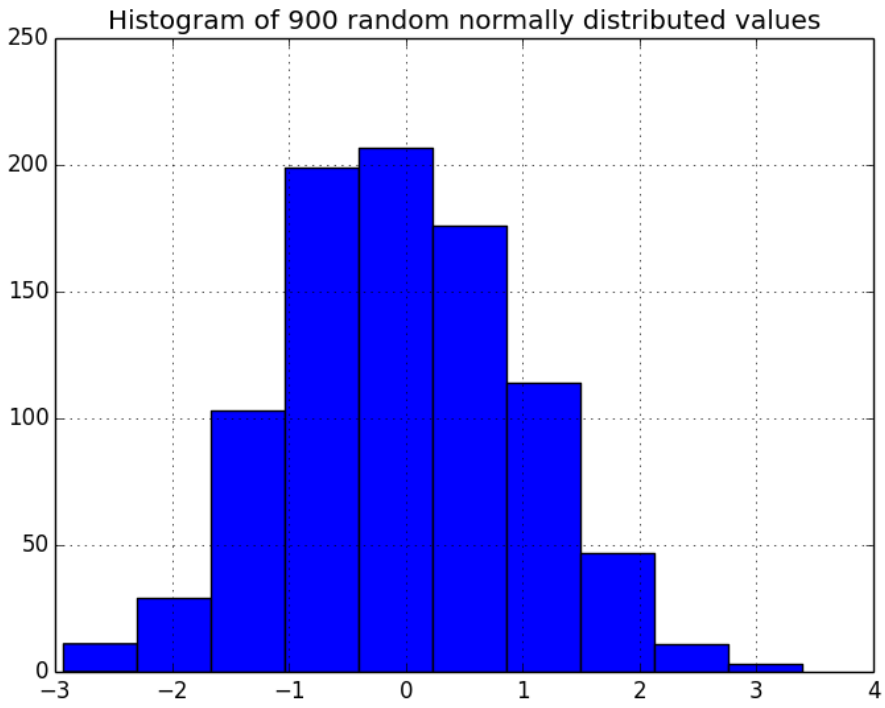

2.0
1.5
1.0
0.5
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0

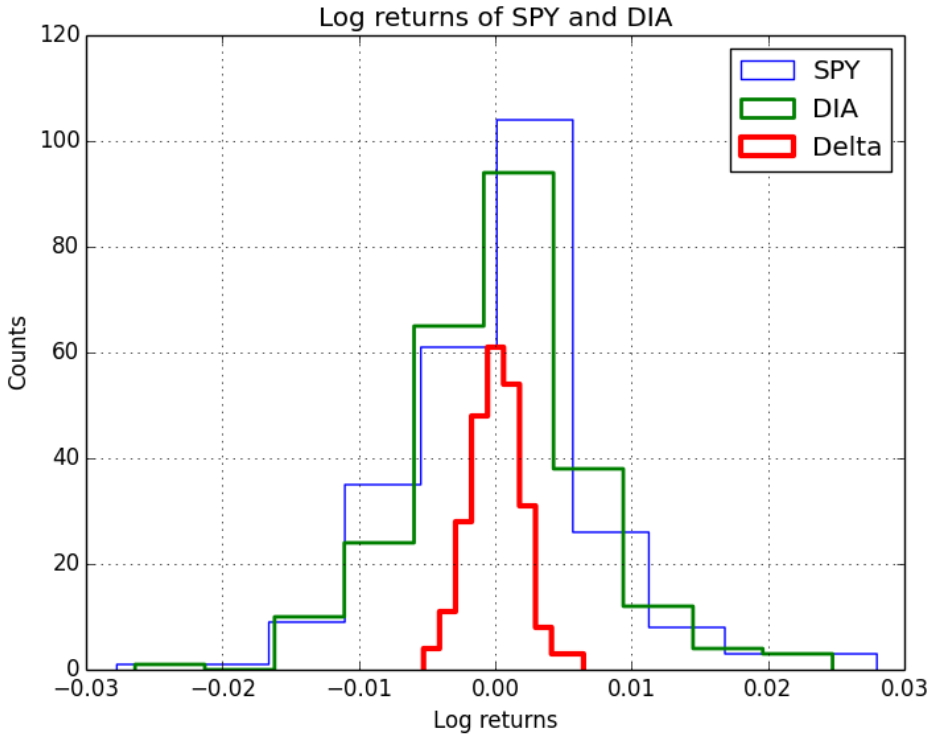

1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0

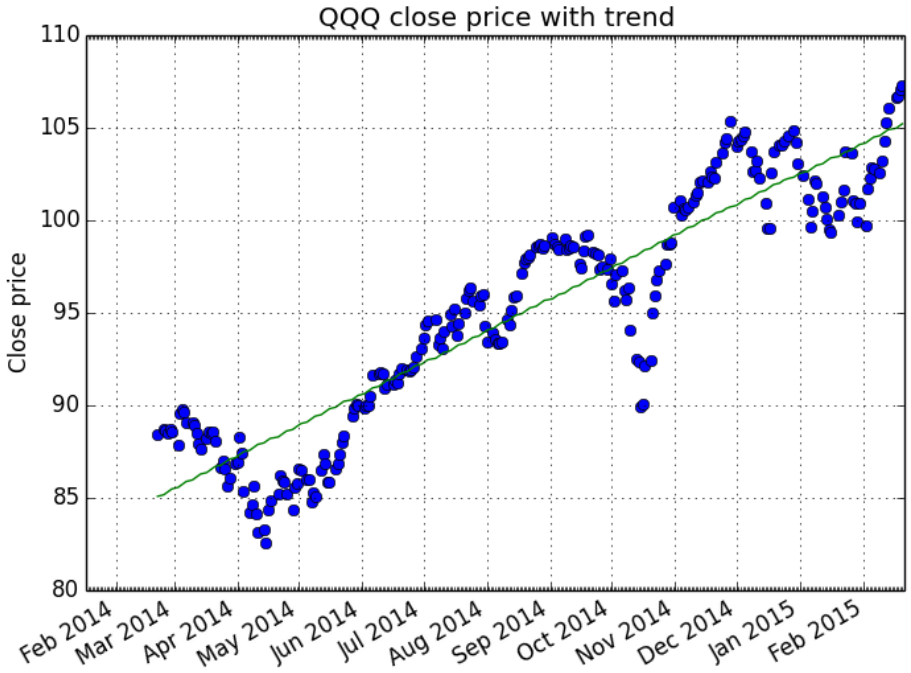


# 10

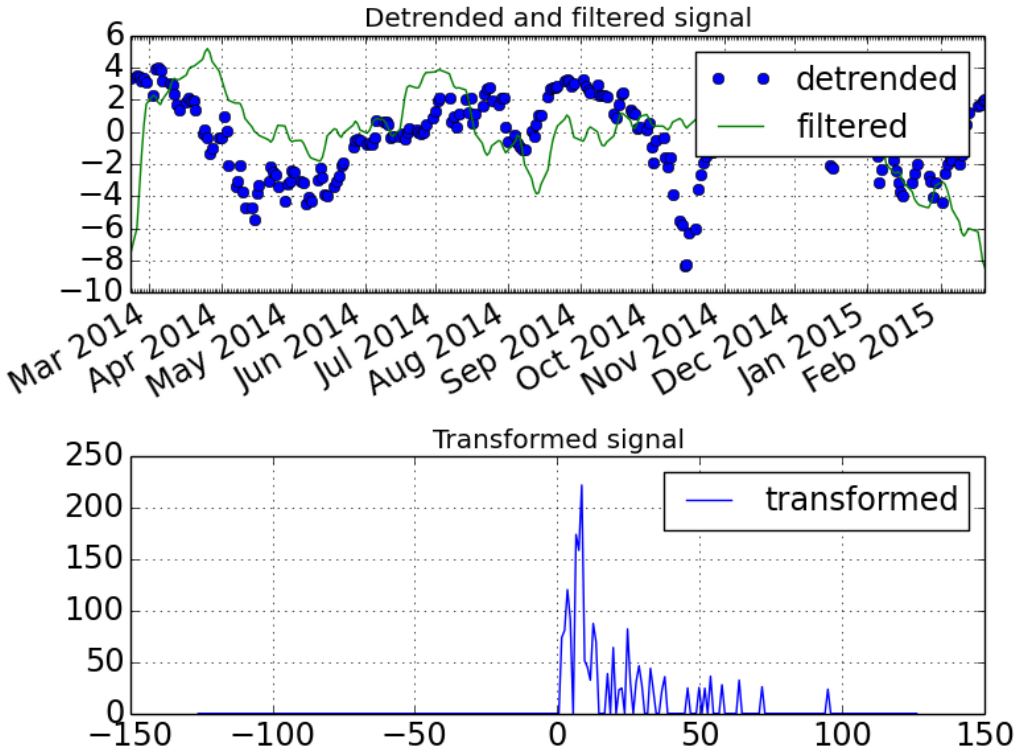
# When NumPy is Not Enough – SciPy and Beyond

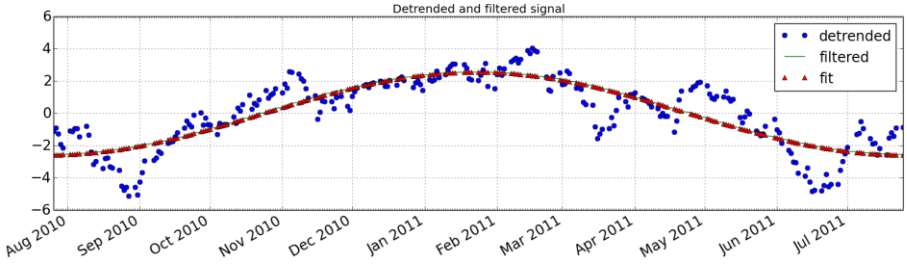


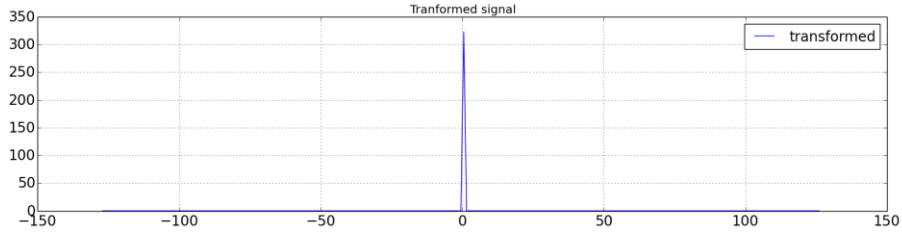

Log returns of SPY and DIA
SPY
DIA
Delta
Counts
Log returns
120
100
80
60
40
20
0
−0.03
−0.02
−0.01
0.00
0.01
0.02
0.03



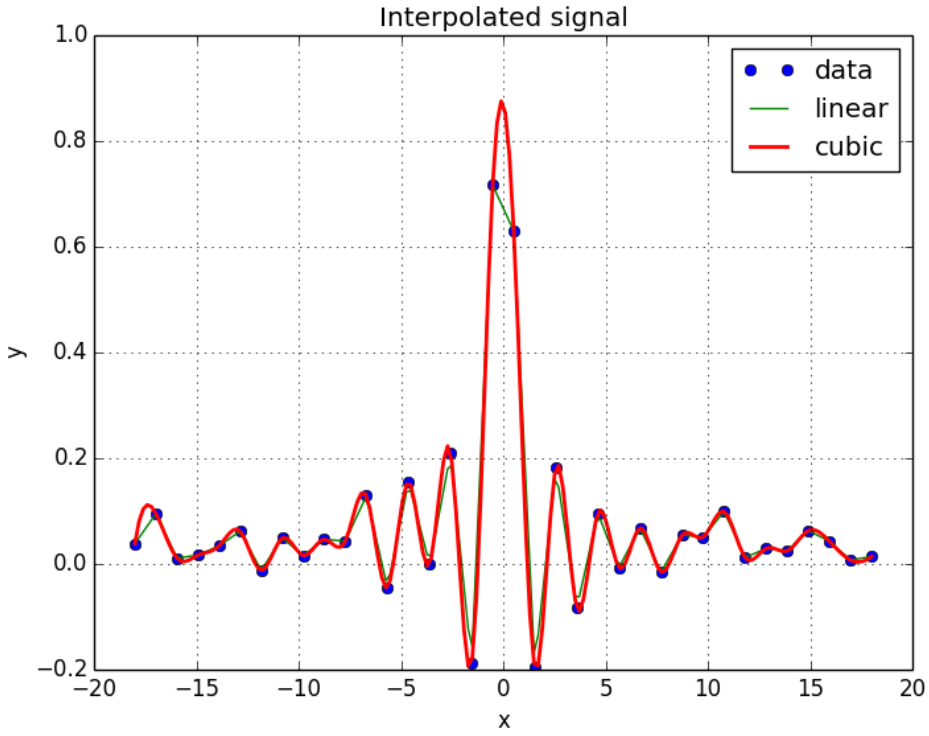

$$\sum_{t=-\infty}^{\infty} x[t]e^{-i\omega t}$$


Detrended and filtered signal
detrended
filtered
6
4
2
0
−2
−4
−6
−8
−10
Mar 2014
Apr 2014
May 2014
Jun 2014
Jul 2014
Aug 2014
Sep 2014
Oct 2014
Nov 2014
Dec 2014
Jan 2015
Feb 2015
Transformed signal
transformed
250
200
150
100
50
0
−150
−100
−50
0
50
100
150




$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}$$

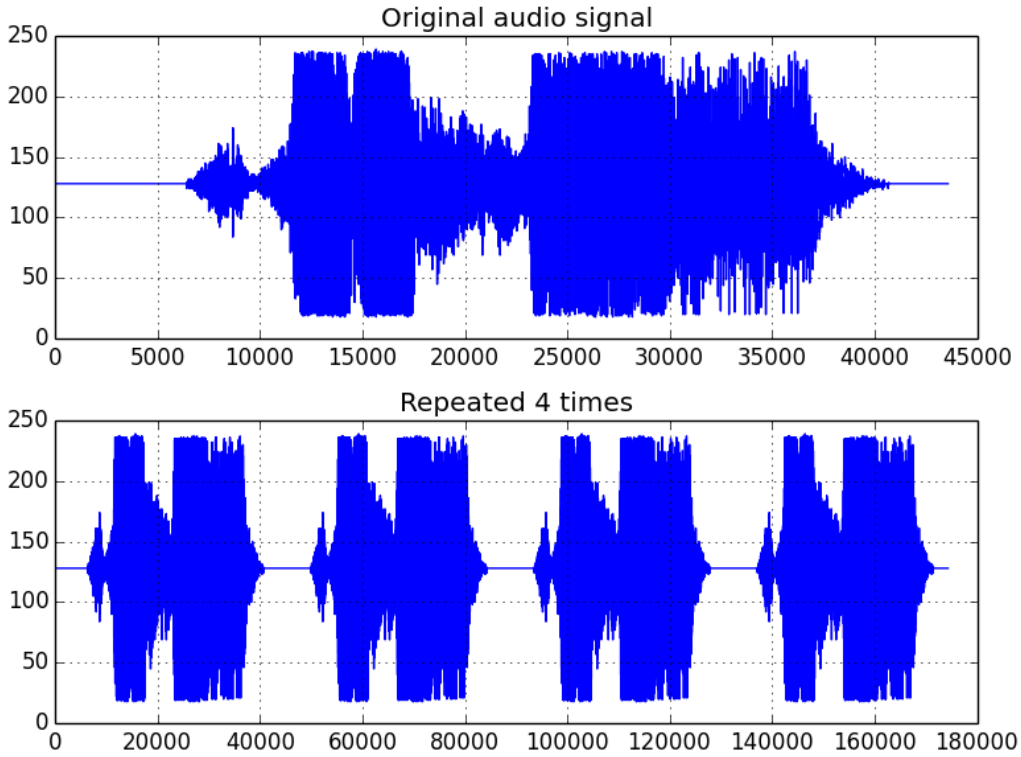

Interpolated signal
data
linear
cubic
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
y
−20
−15
−10
−5
0
5
10
15
20
x

Original Image

Median Filter

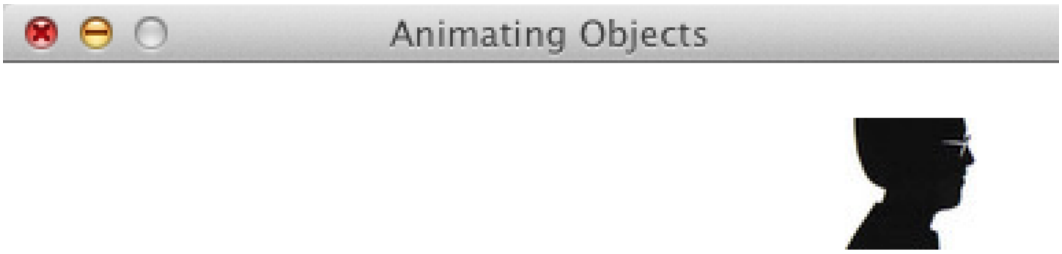
Rotated

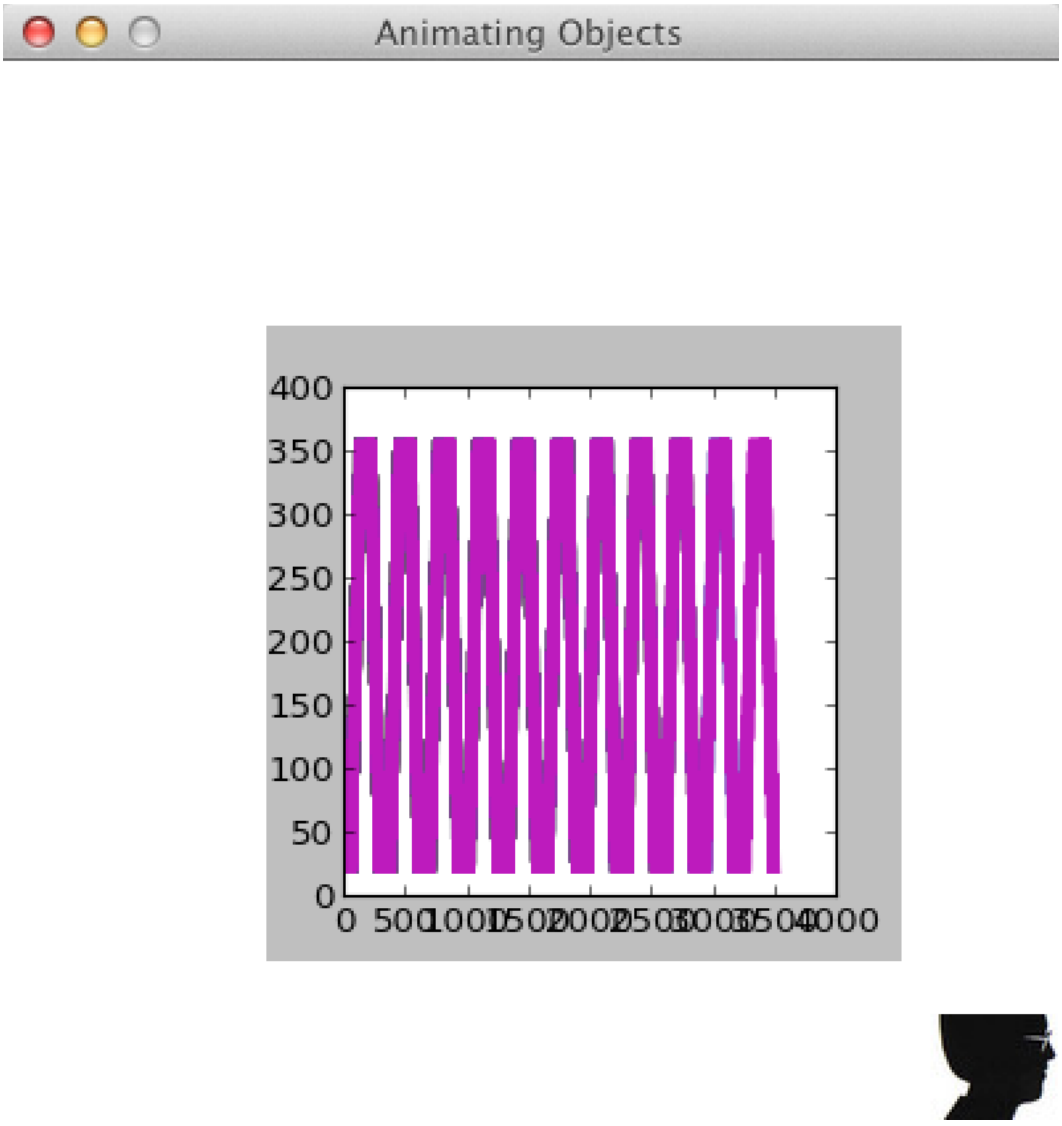
Prewitt Filter

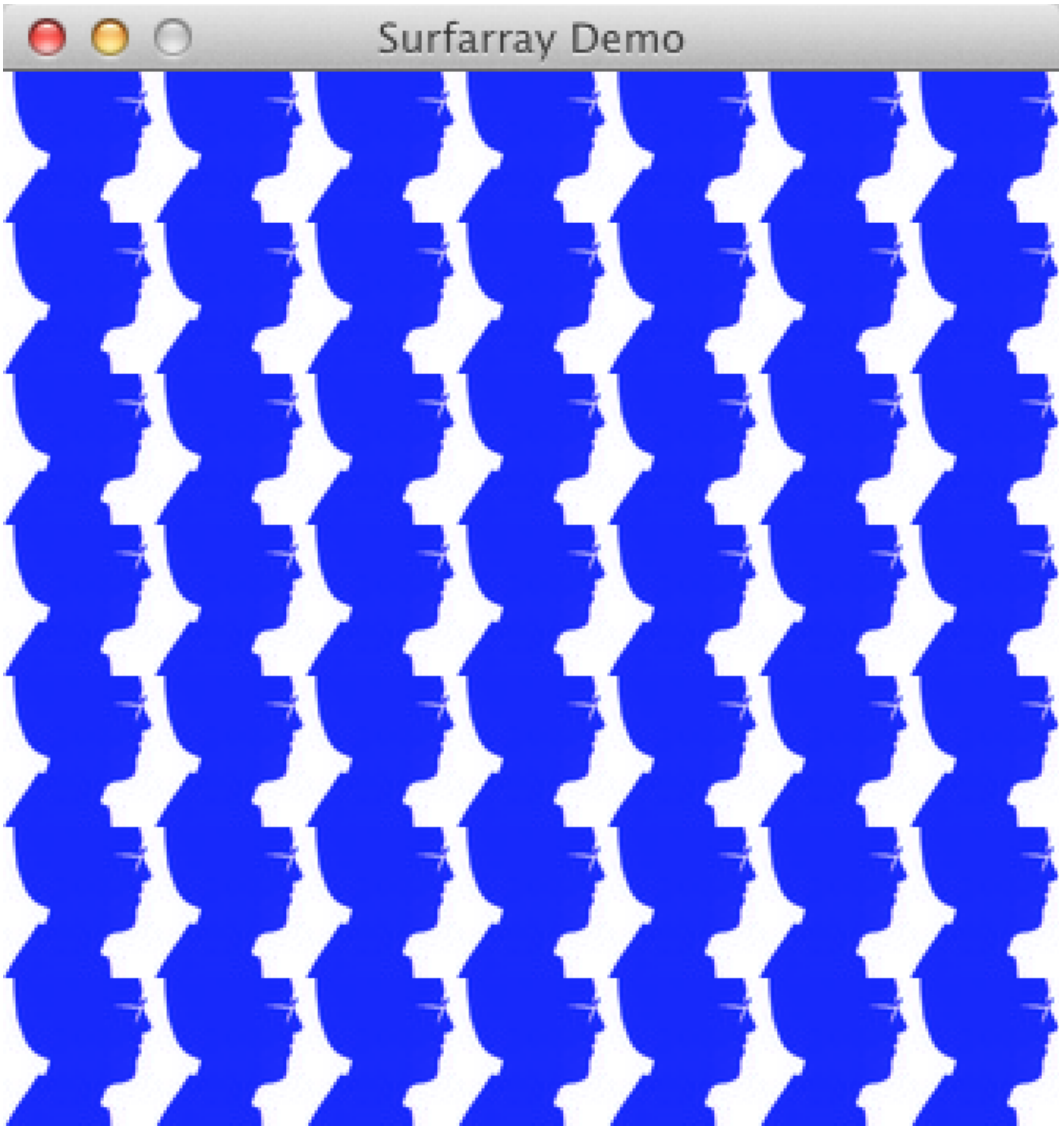

Original audio signal
250
200
150
100
50
0
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
Repeated 4 times
250
200
150
100
50
0
0
20000
40000
60000
80000
100000
120000
140000
160000
180000

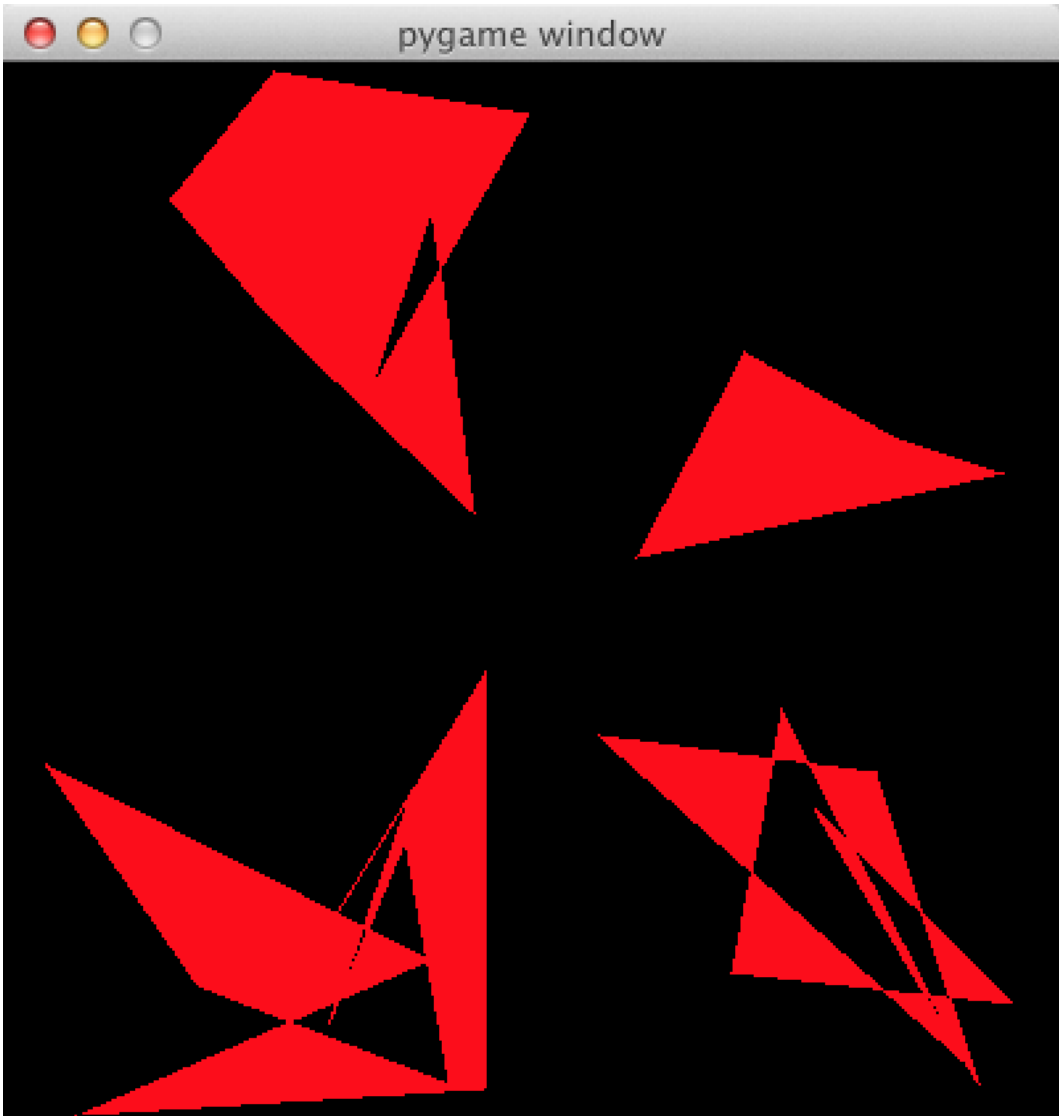
# 11

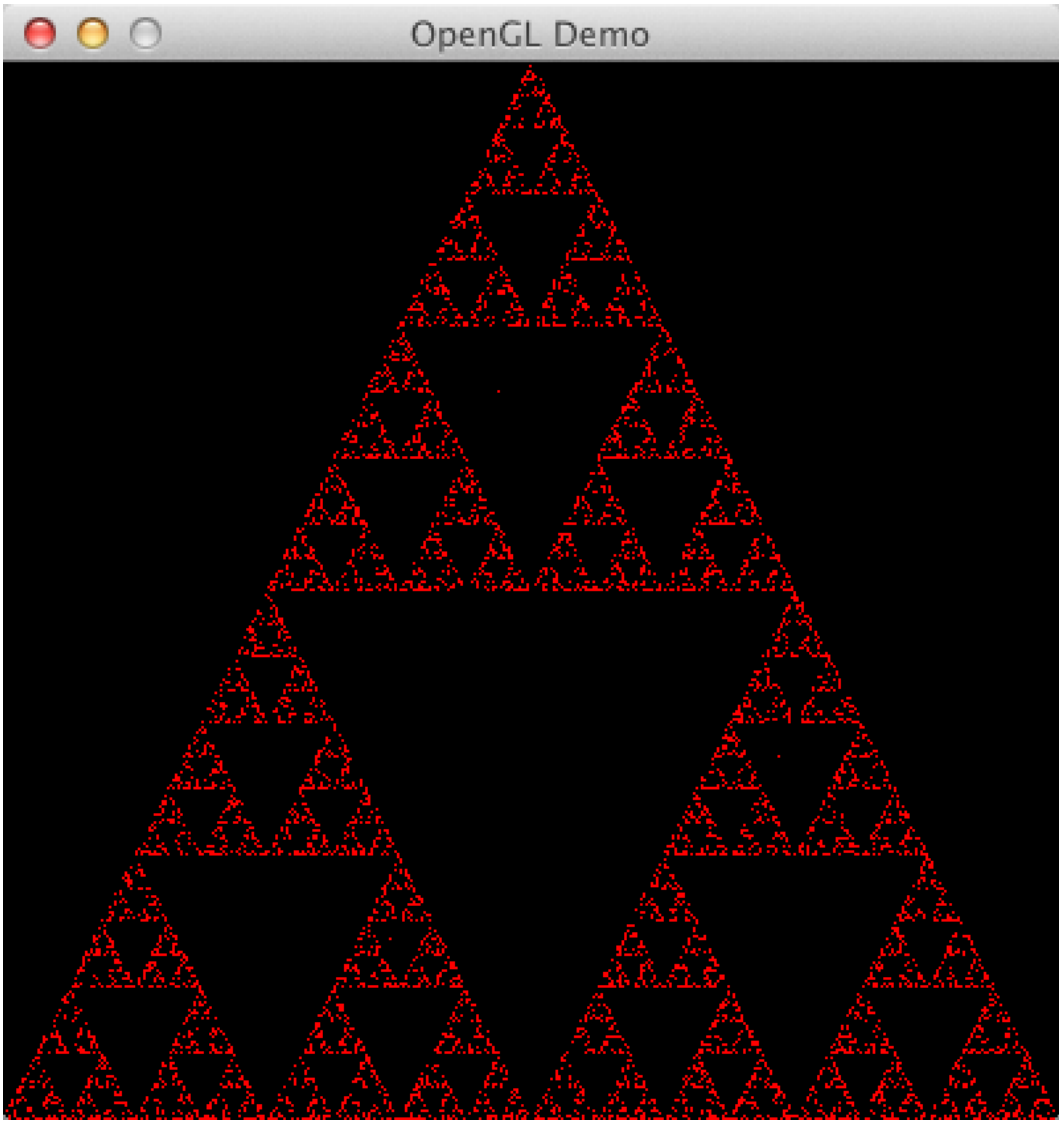
# Playing with Pygame

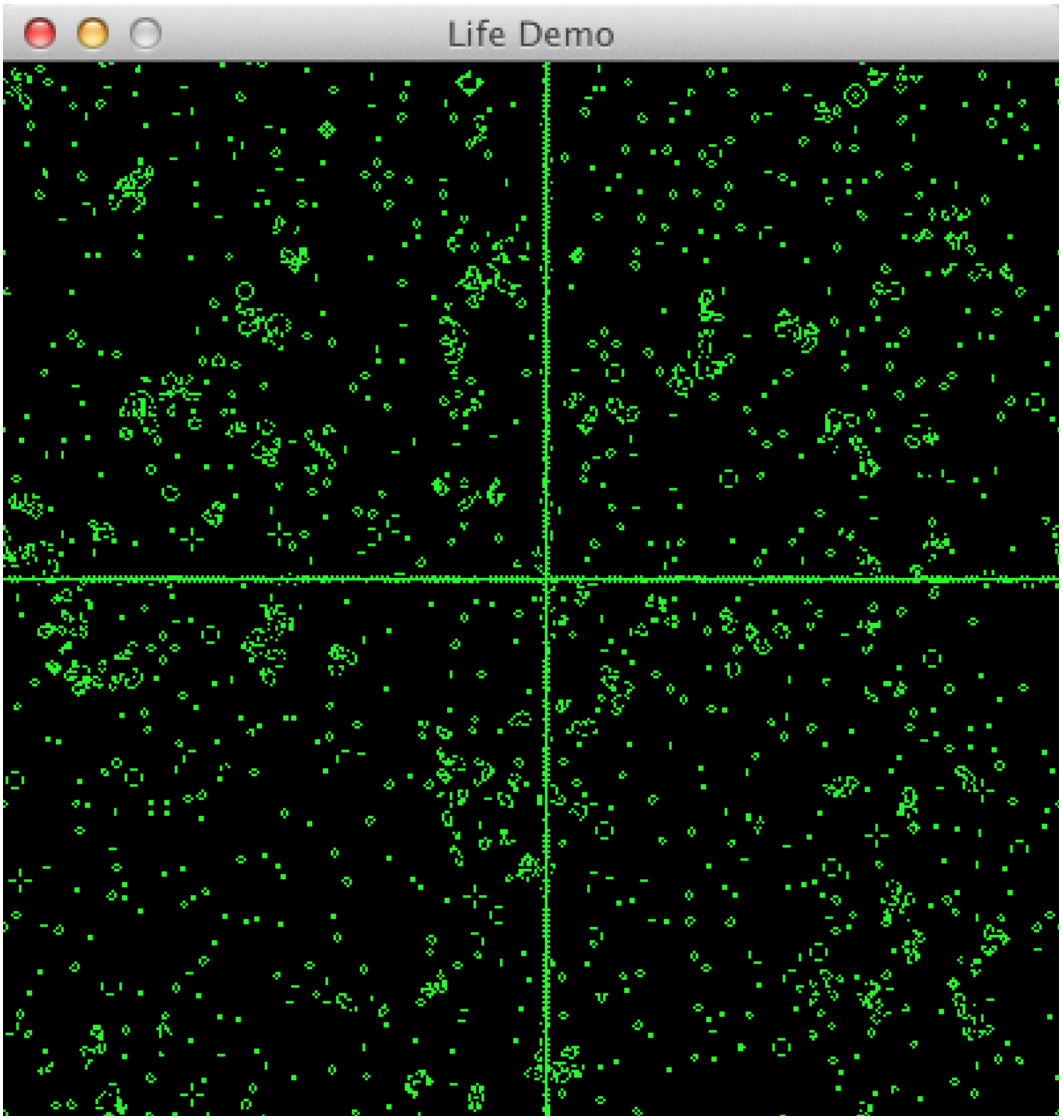

Animating Objects


Animating Objects
400
350
300
250
200
150
100
50
0
0 500 1000 1500 2000 2500 3000 3500 4000


Surfarray Demo


pygame window


OpenGL Demo


Life Demo