
volest

Learn how to apply this code to your own options trading

Getting Started With Python for Quant Finance is the cohort-based course and community that will take you from complete beginner to up and running with Python for quant finance in 30 days.

A complete set of volatility estimators based on Euan Sinclair's Volatility Trading.

<http://www.amazon.com/gp/product/0470181990/tag=quantfincea-20>

The original version incorporated network data acquisition from Yahoo! Finance from `pandas_datareader`. Yahoo! changed their API and broke `pandas_datareader`.

The changes allow you to specify your own data so you're not tied into equity data from Yahoo! finance. If you're still using equity data, just download a CSV from finance.yahoo.com and use the `data.yahoo_data_helper` method to form the data properly.

Volatility estimators include:

- Garman Klass
- Hodges Tompkins
- Parkinson
- Rogers Satchell
- Yang Zhang
- Standard Deviation

Also includes

- Skew
- Kurtosis
- Correlation

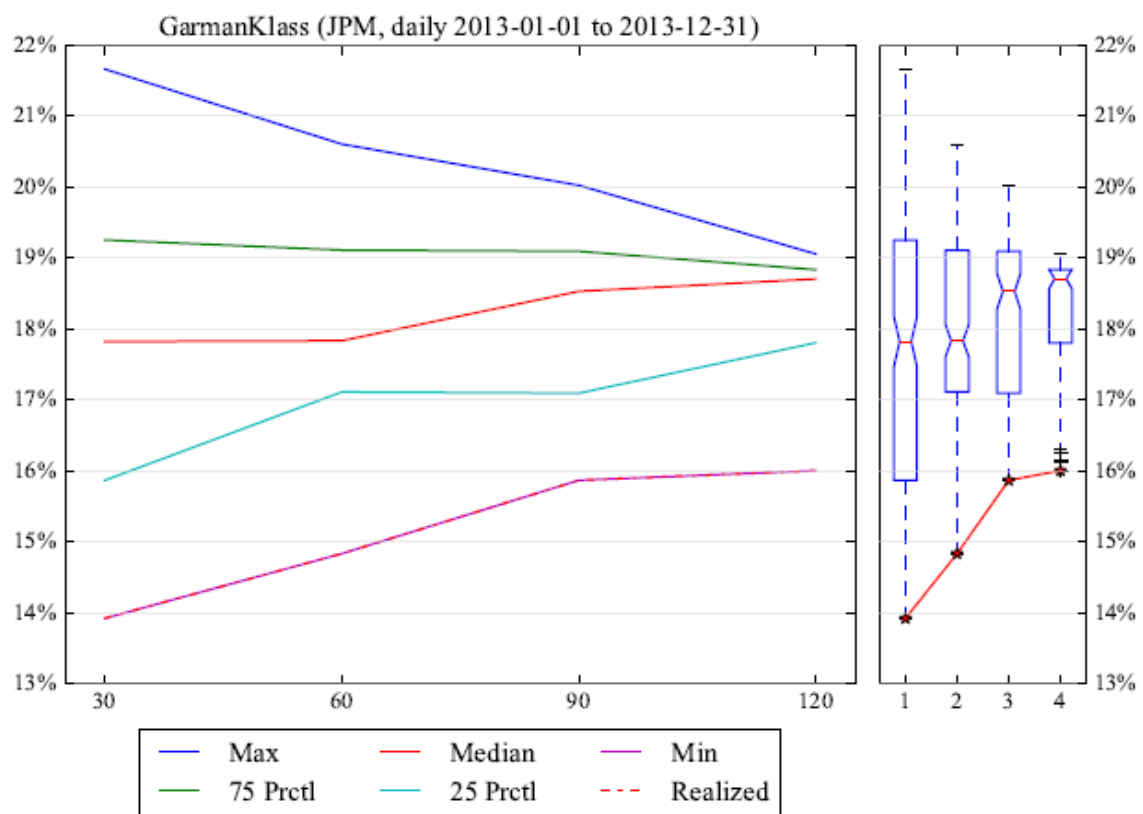
For each of the estimators, plot:

- Probability cones
- Rolling quantiles
- Rolling extremes
- Rolling descriptive statistics
- Histogram

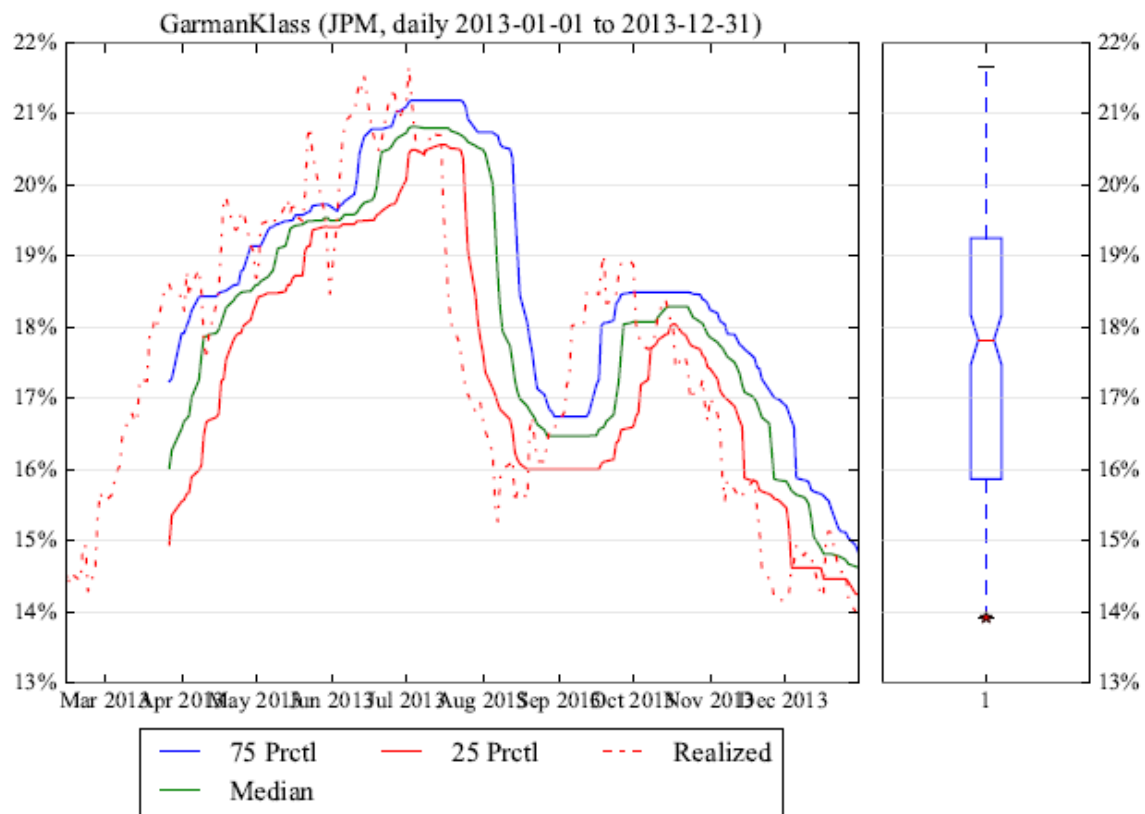
- Comparison against arbitrary comparable
- Correlation against arbitrary comparable
- Regression against arbitrary comparable

Create a term sheet with all the metrics printed to a PDF.

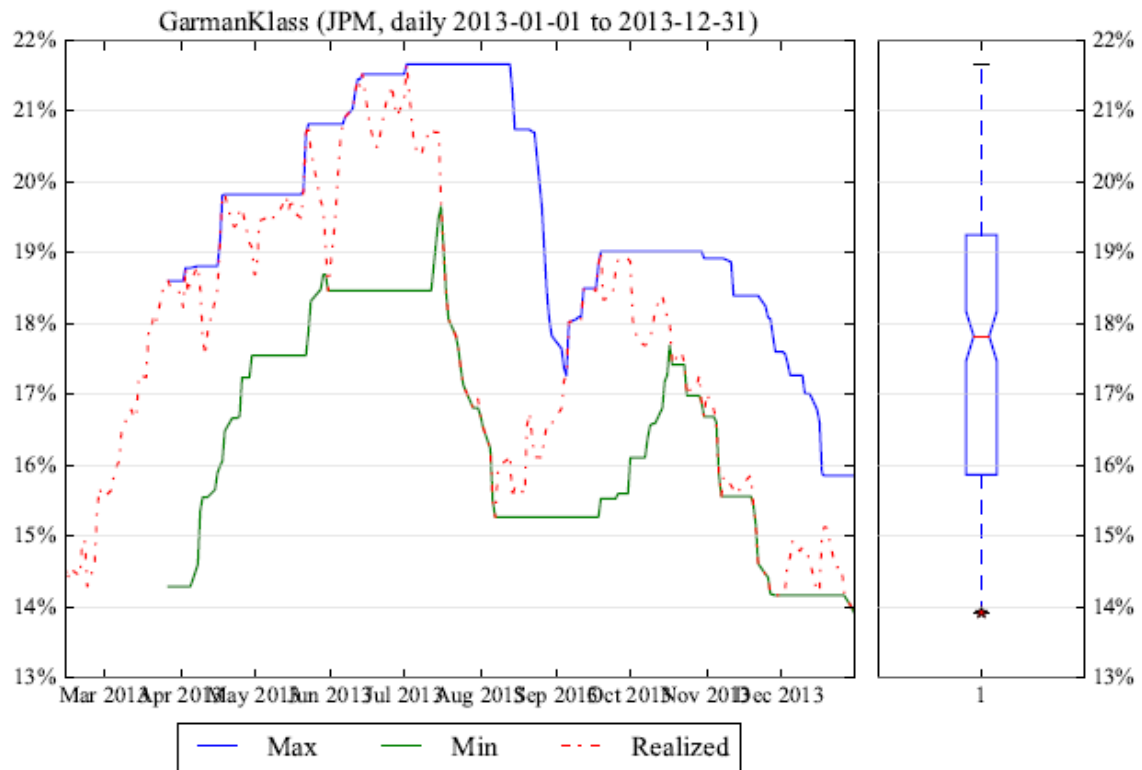
Page 1 - Volatility cones



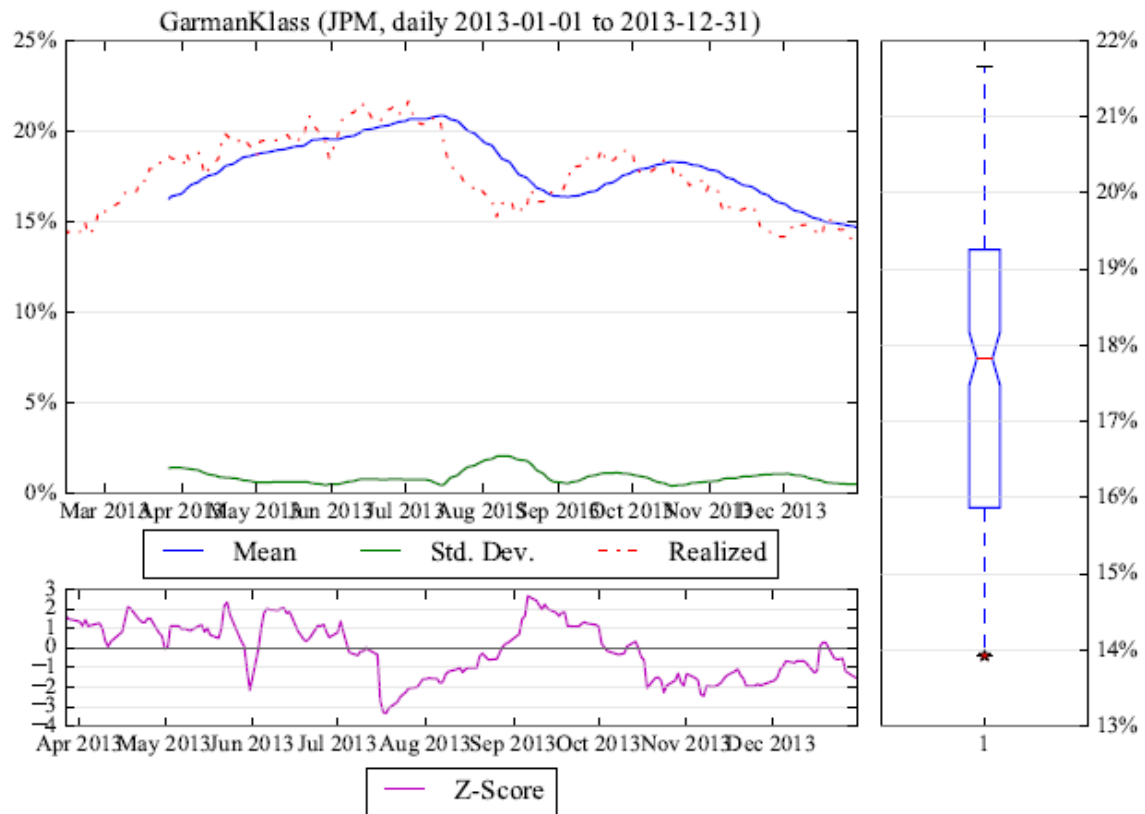
Page 2 - Volatility rolling percentiles



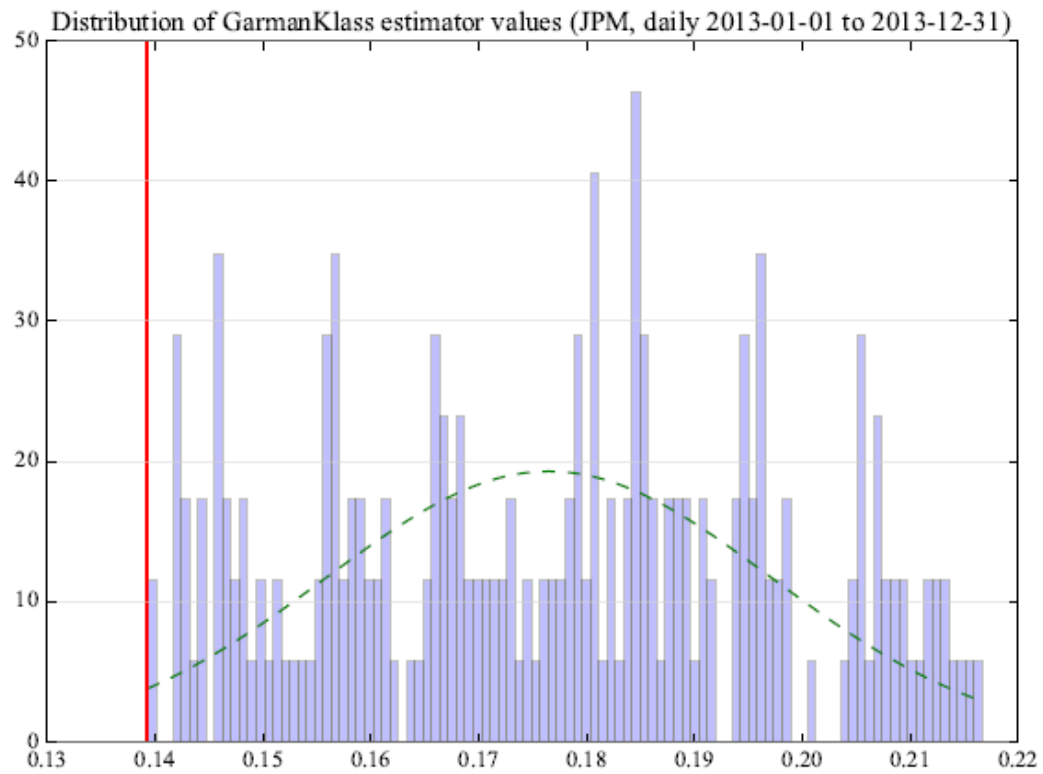
Page 3 - Volatility rolling min and max



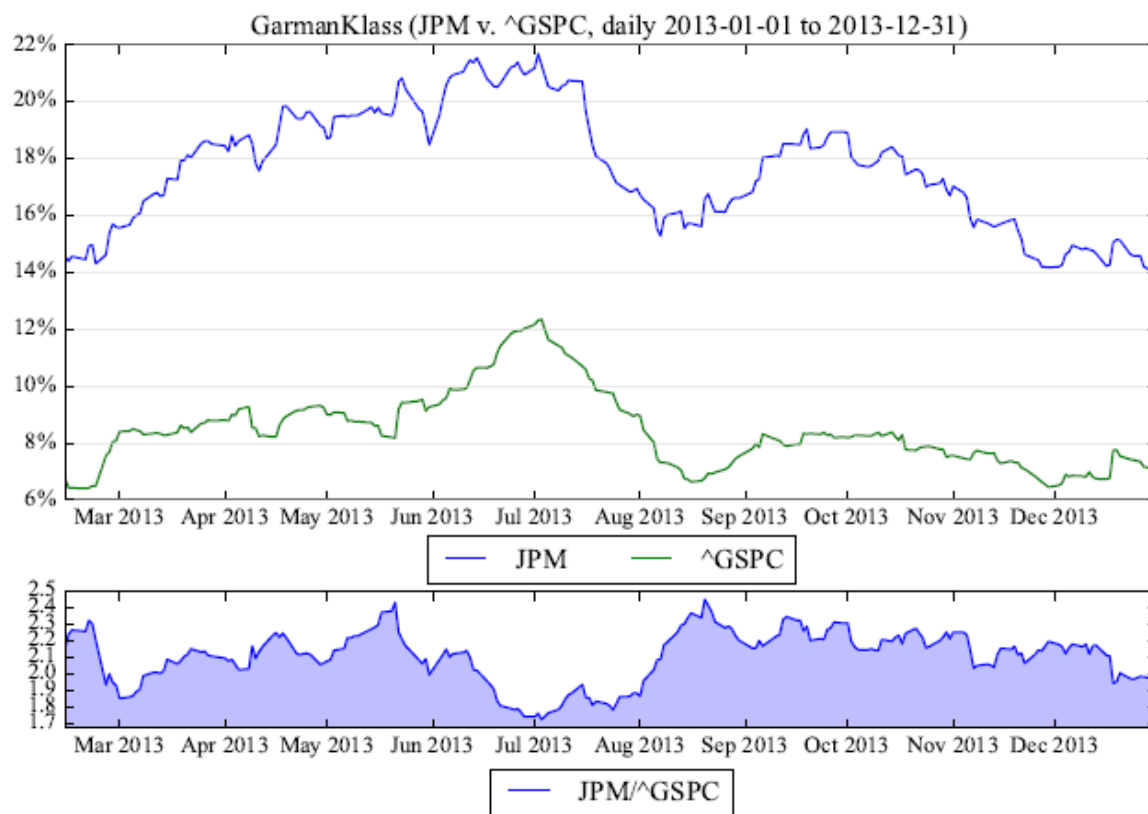
Page 4 - Volatility rolling mean, standard deviation and zscore



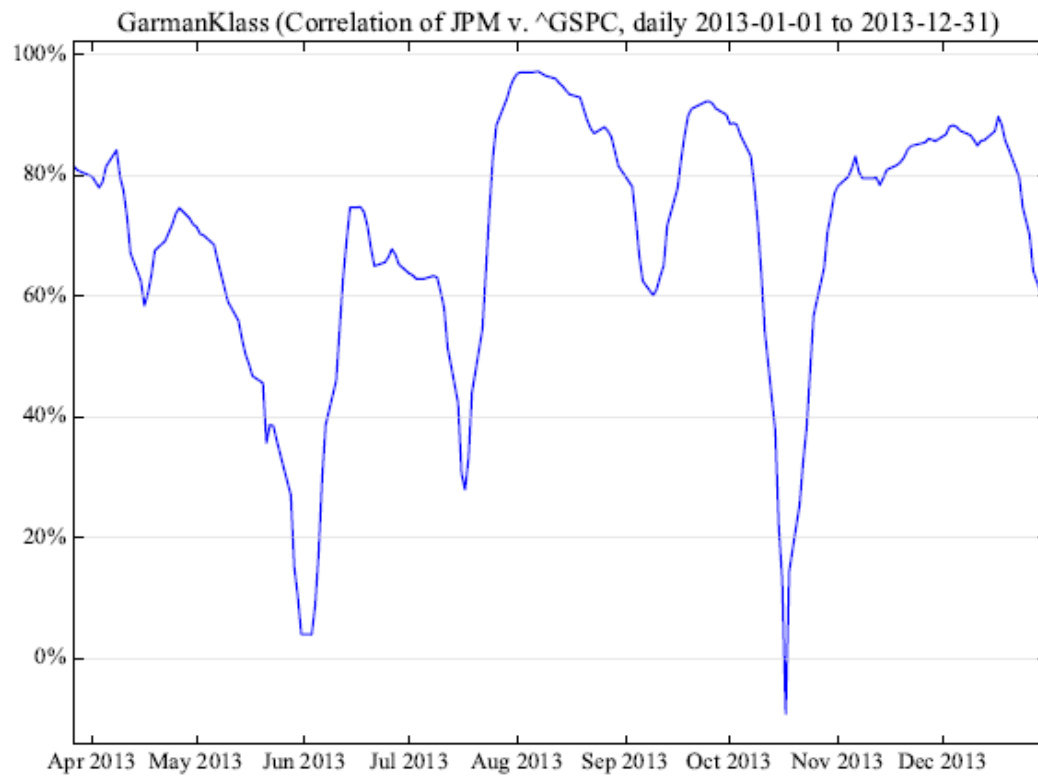
Page 5 - Volatility distribution



Page 6 - Volatility, benchmark volatility and ratio



Page 7 - Volatility rolling correlation with benchmark



Page 3 - Volatility OLS results

OLS Regression Results						
=====						
Dep. Variable:	y	R-squared:	0.931			
Model:	OLS	Adj. R-squared:	0.931			
Method:	Least Squares	F-statistic:	9827.			
Date:	Wed, 19 Jul 2017	Prob (F-statistic):	0.00			
Time:	13:25:00	Log-Likelihood:	1131.7			
No. Observations:	726	AIC:	-2261.			
Df Residuals:	725	BIC:	-2257.			
Df Model:	1					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

x1	1.8734	0.019	99.130	0.000	1.836	1.911
=====						
Omnibus:	270.169	Durbin-Watson:	0.090			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1264.273			
Skew:	1.647	Prob(JB):	2.93e-275			
Kurtosis:	8.563	Cond. No.	1.00			

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Example usage:

```
1
2 from volatility import volest
3 import yfinance as yf
4
5 # data
6 symbol = 'JPM'
7 bench = 'SPY'
8 estimator = 'GarmanKlass'
9
10 # estimator windows
11 window = 30
12 windows = [30, 60, 90, 120]
13 quantiles = [0.25, 0.75]
14 bins = 100
15 normed = True
16
17 # use the yahoo helper to correctly format data from finance.yahoo.com
18 jpm_price_data = yf.Ticker(symbol).history(period="5y")
19 jpm_price_data.symbol = symbol
20 spx_price_data = yf.Ticker(bench).history(period="5y")
21 spx_price_data.symbol = bench
22
```

```
23 # initialize class
24 vol = volest.VolatilityEstimator(
25     price_data=jpm_price_data,
26     estimator=estimator,
27     bench_data=spx_price_data
28 )
29
30 # call plt.show() on any of the below...
31 _, plt = vol.cones(windows=windows, quantiles=quantiles)
32 _, plt = vol.rolling_quantiles(window=window, quantiles=quantiles)
33 _, plt = vol.rolling_extremes(window=window)
34 _, plt = vol.rolling_descriptives(window=window)
35 _, plt = vol.histogram(window=window, bins=bins, normed=normed)
36
37 _, plt = vol.benchmark_compare(window=window)
38 _, plt = vol.benchmark_correlation(window=window)
39
40 # ... or create a pdf term sheet with all metrics in term-sheets/
41 vol.term_sheet(
42     window,
43     windows,
44     quantiles,
45     bins,
46     normed
47 )
```

Hit me on twitter with comments, questions, issues @jasonstrimpel