**Estimating Financial Risk through Monte Carlo Simulation**

Modeling Value at Risk (VaR) with Linear Regression Under Normal Distribution Assumption

# **Outline**

* What Are We Getting Into?
* Basic Terms
* Monte Carlo Risk Modeling
* Results / Evaluations

# **What Are We Getting Into?**

* Train a linear regression model on stock data
* Calculate the risk by running the trained model on virtual markets produced by Monte Carlo Simulation
* We will assume normal distribution for features (market factors) and use multivariate normal distribution for the simulation
* Monte Carlo Simulation is massively parellelizable and Spark is very useful for this!

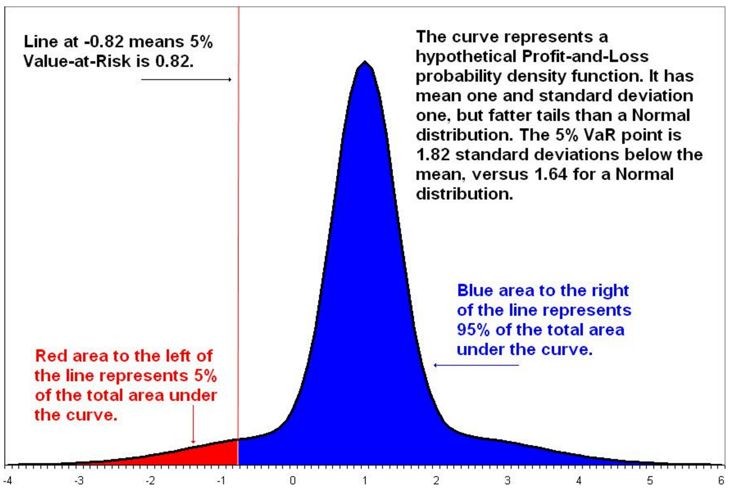


# **Basic Terms**

1. Value at Risk (VaR)

A simple measure of investment risk that tries to provide a reasonable estimate of maximum probable loss in value of an investment over the particular period

e.g.) A VaR of 1 mil dollars with a 5% p-value and two weeks -> your investment stands 5% chance of losing more than 1 mil dollars over two weeks



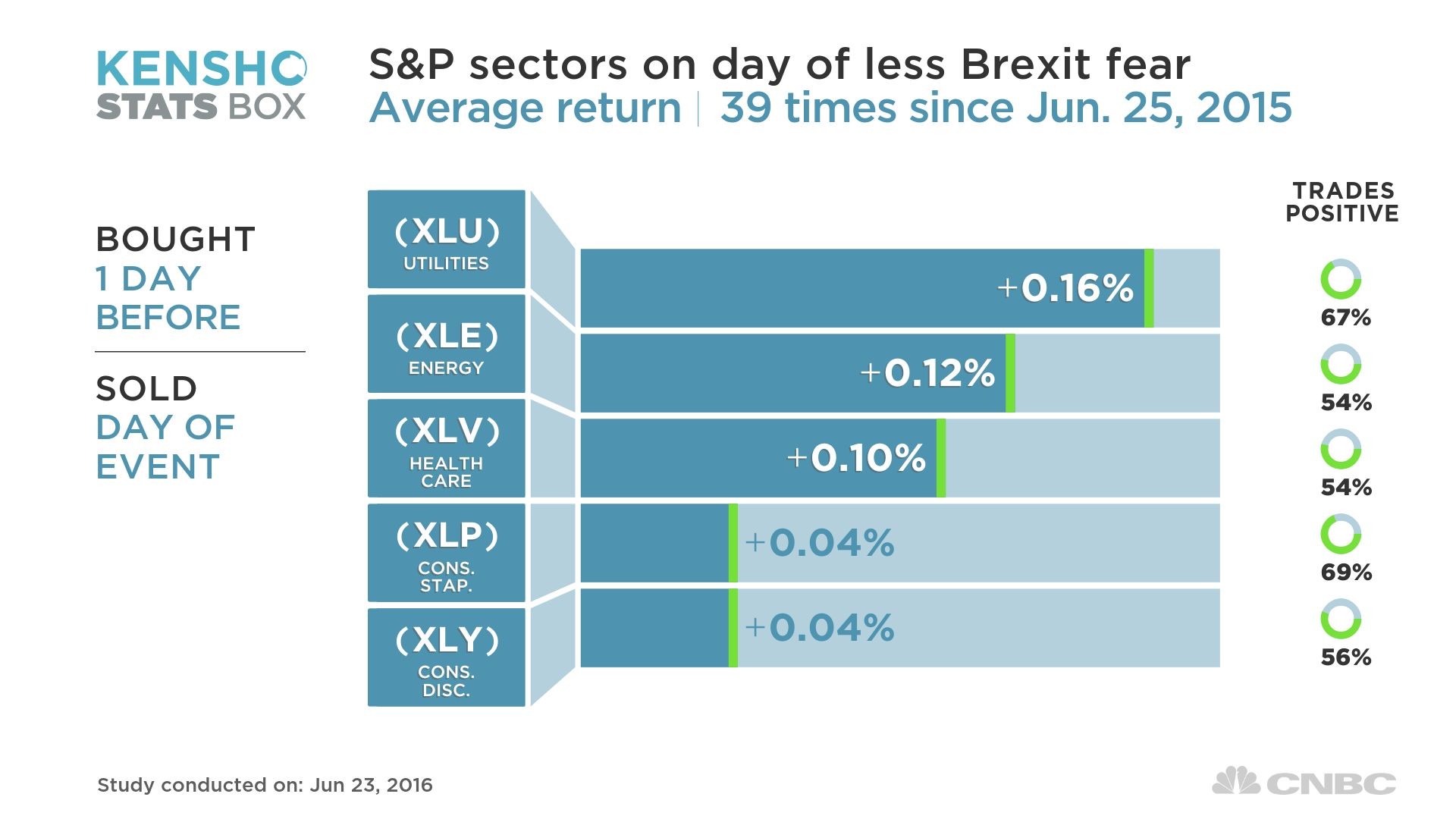
# **Basic Terms**

1. 5% VaR

# **Basic Terms**

1. Conditional Value at Risk (CVaR) Expected Shortfall (average of VaR values)

e.g.) A CVaR of 5 million dollars with a 5% q-value and two weeks indicates the belief that the average loss in the worst 5% of outcomes is 5 million dollars.



# **Basic Terms**

1. Market Factors

A value that can be used as an indicator of macro aspects of the financial climate at a particular time

# **Basic Terms**

1. Resilient Distributed Datasets (RDDs)

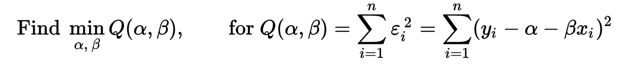
Spark revolves around the concept of a resilient distributed dataset (RDD), which is a fault-tolerant collection of elements that can be operated on in parallel.

# **Basic Terms**

1. Resilient Distributed Datasets (RDDs)

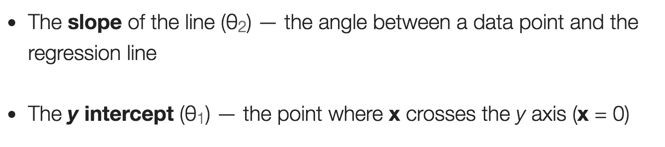
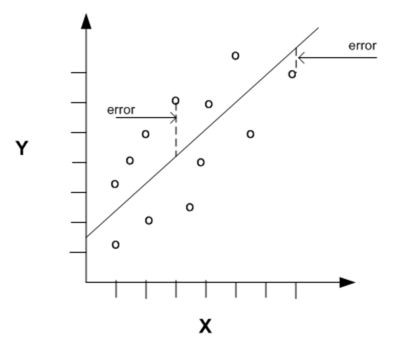
It is an immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster.

RDDs can contain any type of Python, Java, or Scala objects, including user-defined classes.



# **Basic Terms**

1. Linear Regression
   * Try to fit the model with a linear assumption
   * Find parameters which minimize errors



# **Basic Terms**

1. Linear Regression

# **Basic Terms**

1. Monte Carlo Simulation

Monte Carlo simulation performs risk analysis by building models of possible results by substituting a range of values—a probability distribution—for any factor that has inherent uncertainty. It then calculates results over and over, each time using a different set of random values from the probability functions.

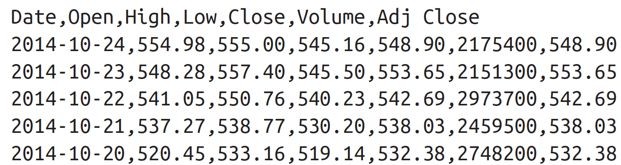
# **Methods for Calculating VaR**

1. Variance-Covariance
2. Historical Simulation
3. Monte Carlo Simulation

# **Monte Carlo Risk Modeling**

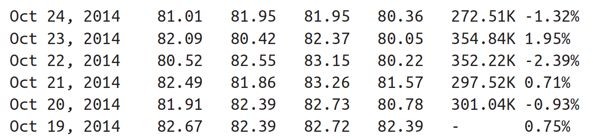
Our Approach

* + Time interval: two weeks
  + Model: Linear Regression
  + Features (x): four market factors
  + Dataset (y): historical data of 3,000 stocks. Returns (change of stock values)
  + Objective: Calculate VaR and CVaR of stocks with Monte Carlo Simulation



# **Dataset**

* + Stock History Data from Yahoo (GOOGL.csv)



# **Dataset**

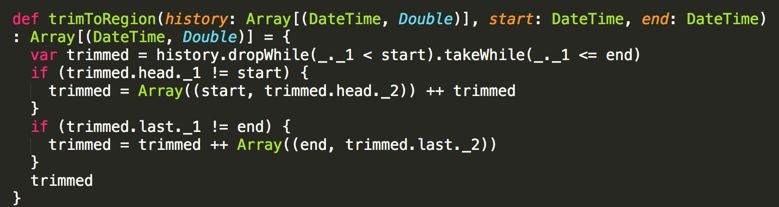
* + Stock History Data from investing.com (CrudeOil.tsv)



# **Preprocessing**

* + Data Point Generation (Two-week interval)

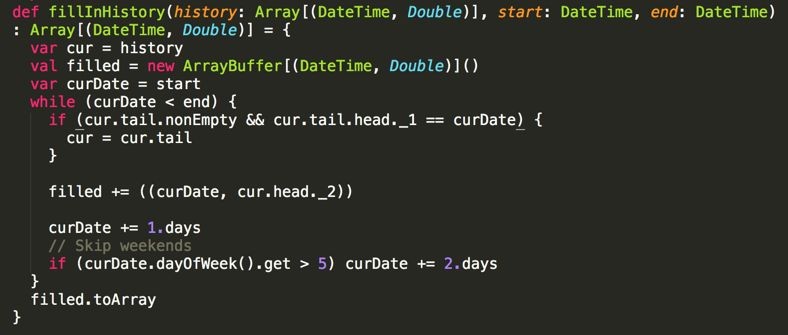
(price on day A - price 14 days later [= 10 rows below]) / (price on day A)



# **Preprocessing**

* + Trimming Data Matrix (no need for details)

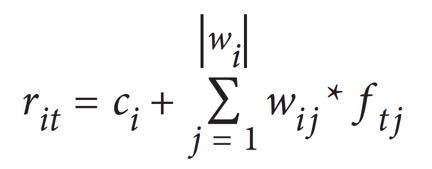
Set the start date and the end date for factors/stocks



# **Preprocessing**

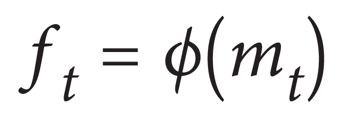
* + Trimming Data Matrix (no need for details)

Fill in the missing values with the value at the closest date



# **Calculation for Parameters of Linear Regression**

A Monte Carlo risk model typically phrases each instrument’s return (the change of stock price over a time period) in terms of a set of market factors.



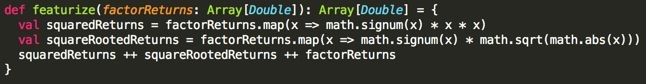
# **Calculation for Parameters of Linear Regression**

Feature Vector with Market Factors

* + NASDAQ

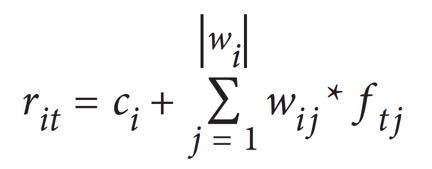
- S&P 500

* + Crude Oil Price
  + US 30-year Treasury Bonds



# **Calculation for Parameters of Linear Regression**

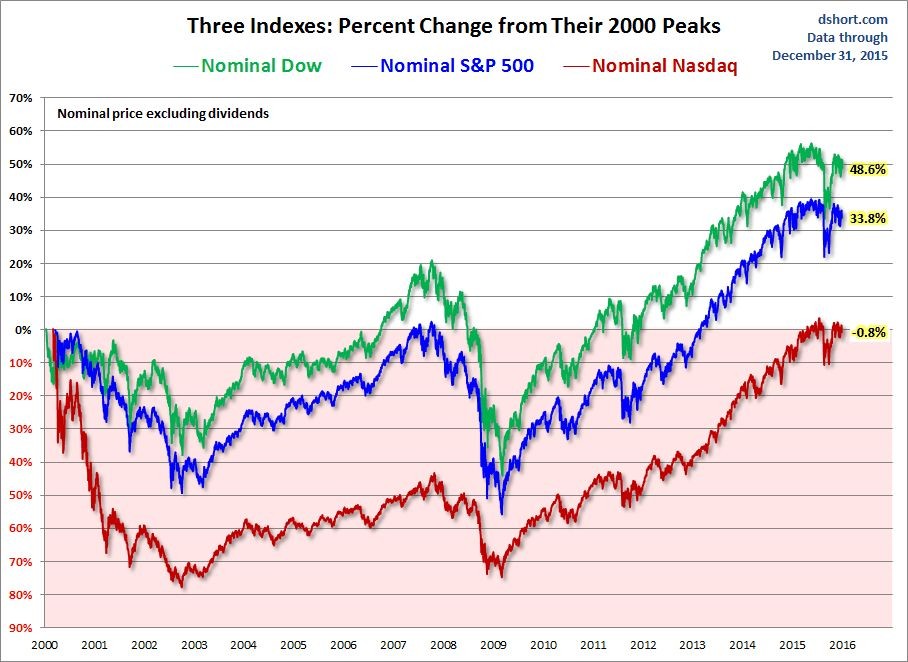
Feature vector from the sample code (x: stock value change, sign of the value is preserved)



# **Calculation for Parameters of Linear Regression**

Linear Regression Model

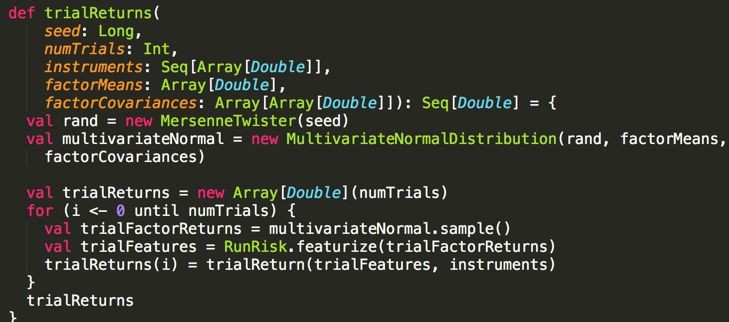
w: weights for features, f: feature, c: intercept, r: return, r: return, i: stock, j: feature factor, t: trials



# **Monte Carlo Simulation**

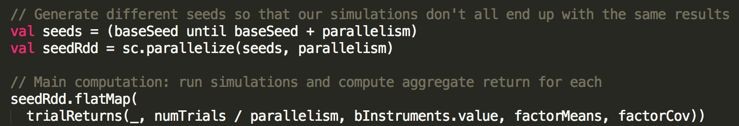
* + Calculate Covariance matrix of four market factors

Closer to the reality! (comparing to independence assumptions)



# **Monte Carlo Simulation**

* + Generate samples of market factor values following multivariate normal distribution



# **Parallel Computations with RDDs**

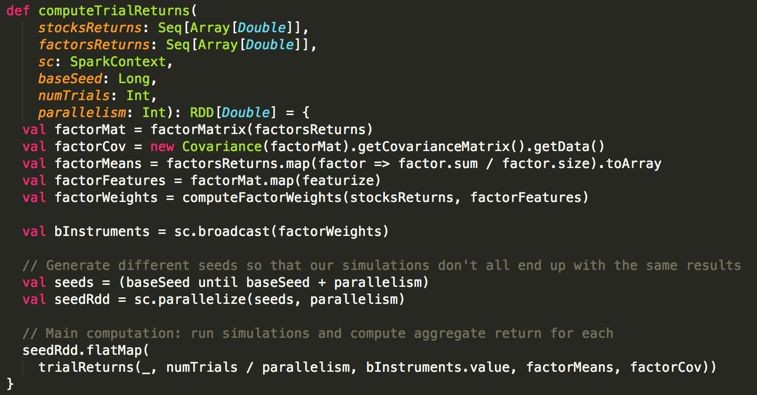
- # of trials: 10,000,000

- # of RDDs: 1,000

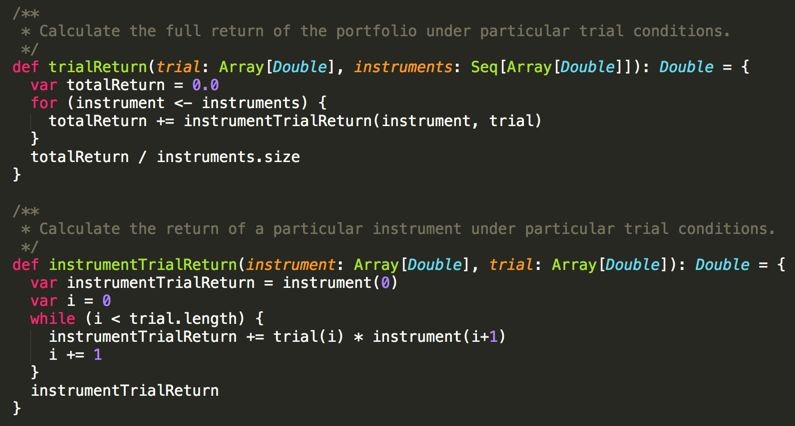
* + Use different seed for Mersenne Twister random generator and feed it to multivariate normal sample for each trial

# **One RDD for One Trial**

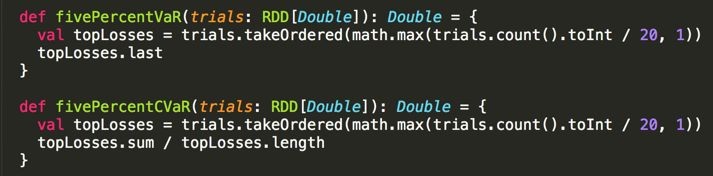
* + One trial simulates one virtual market situation
  + Each market situation is simulated by features sampled by multivariate normal distribution of four market factors and the trained Linear Regression model parameters
  + For each market situation, we calculate the average of VaRs of all stock prices (increase/decrease)



# **One RDD for One Trial**

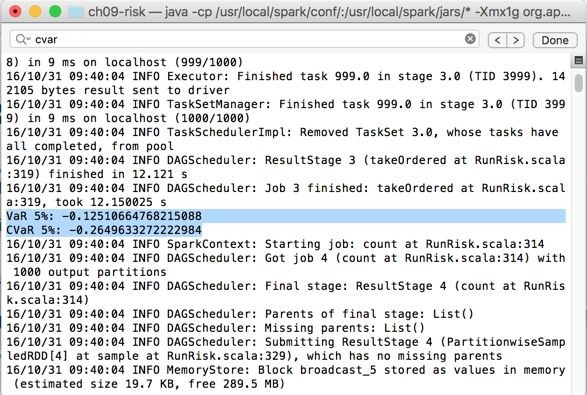


**One RDD for One Trial**



**Finally, VaR and CVaR**

* + Aggregate all trial results



# **Results & Evaluation**

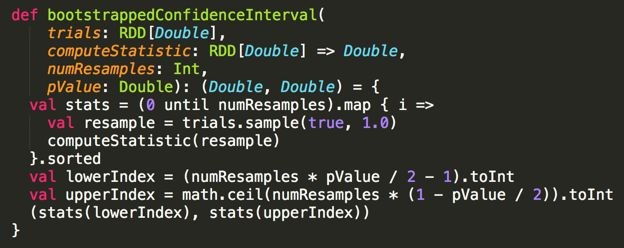
**Results & Evaluation**

* + Confidence Interval (95%)

We are 95% confident to say that the VaR would fall into this interval.

* + Bootstrapping

Resample from the subset of VaRs resulted from trials



# **Results & Evaluation**

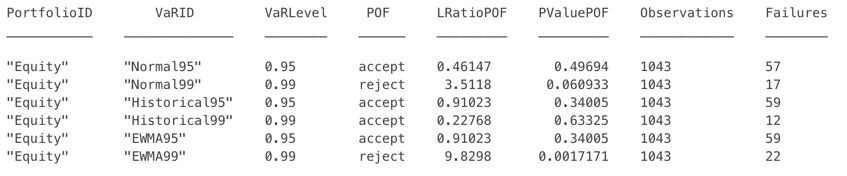
* + Bootstrapped Confidence Interval (95%)

Get the confidence interval from bootstrapped dataset.

# **Results & Evaluation**

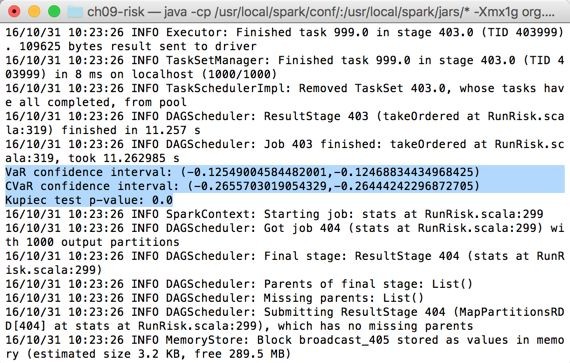
* + Kupiec’s proportion-of-failures (POF) test

Counts the number of times that the losses exceeded the VaR. The null hypothesis is that the VaR is reasonable, and a sufficiently extreme test statistic means that the VaR estimate does not accurately describe the data.



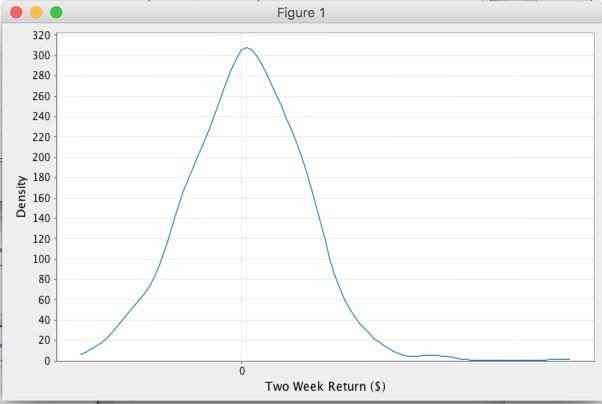
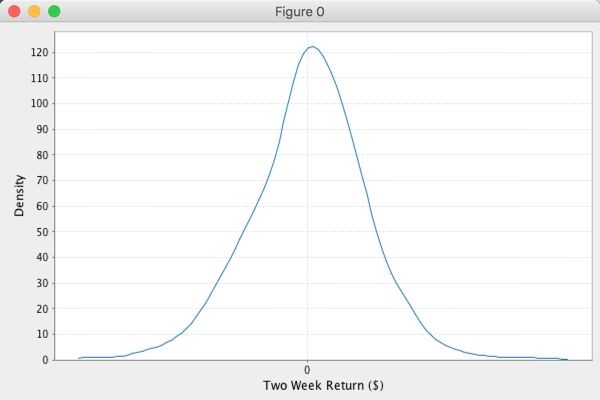
# **Results & Evaluation**

* + Kupiec’s proportion-of-failures (POF) test



# **Results & Evaluation**

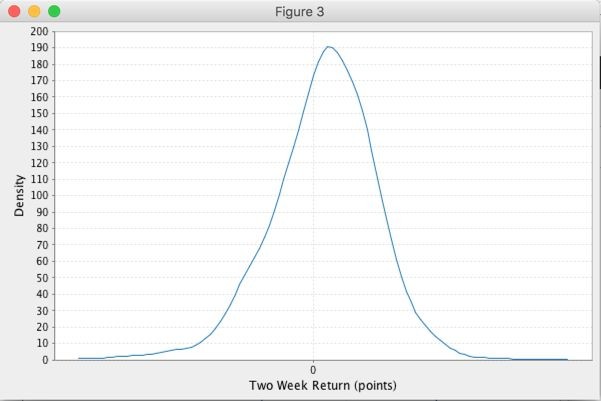
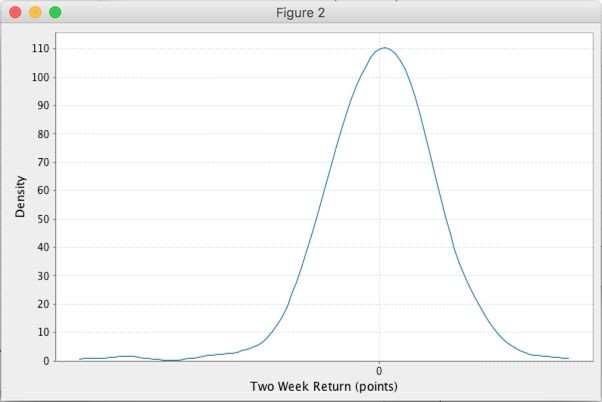
Kupiec test says that this VaR model is not reasonable...



# **Results & Evaluation**

Market Factor Distributions

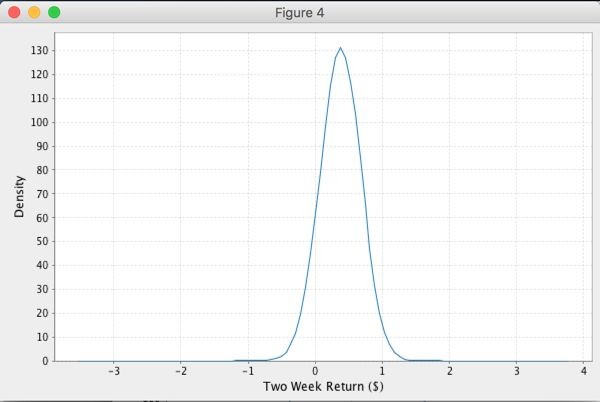
Crude Oil US 30-Year Treasury



# **Results & Evaluation**

Market Factor Distributions

S&P 500 NASDAQ



# **Results & Evaluation**

Monte Carlo Simulation

3,000 stocks

# **References**

<http://spark.apache.org/docs/latest/programming-guide.html>

<https://github.com/sryza/aas>

<https://www.mathworks.com/help/risk/pof.html>

<https://en.wikipedia.org/wiki/Linear_regression>

<http://www.palisade.com/risk/monte_carlo_simulation.asp>

Advanced Analytics with Spark: Patterns for Learning from Data at Scale (2015) - Josh Wills, Sandy Ryza, Sean Owen, and Uri Laserson

# **Image Resources**

<http://sakiicelimbekardas.blogspot.com/2016/02/stock.html>

<http://www.cnbc.com/2016/06/23/sp-500-sectors-in-the-brexit-crosshairs.html>

<http://www.cnbc.com/2015/07/17/5-tech-trades-on-nasdaqs-record-close.html>

[http://www.investing.com/analysis/the-s-p-500,-dow-and-nasdaq-since-their-2000-highs-37](http://www.investing.com/analysis/the-s-p-500%2C-dow-and-nasdaq-since-their-2000-highs-378646) [8646](http://www.investing.com/analysis/the-s-p-500%2C-dow-and-nasdaq-since-their-2000-highs-378646)