OPEN DATA SCIENCE CONFERENCE

#ODSC

@ODSC

Boston | April 30 - May 4, 2019

#ODSC

BOSTON
APR 30 – MAY 3

Modeling Volatility Trading Using Econometrics and Machine Learning in Python

Stephen Lawrence, PhD

Head of Investment Management Fintech Data Science,
Vanguard

#ODSC

BOSTON
APR 30 – MAY 3

Modeling Volatility Trading Using Econometrics and Machine Learning in Python

Eunice Hameyie-Sanon

Sr. Data Scientist - Investment Management Fintech Strategies, Vanguard

Important Information

The opinions and materials presented in this training session are solely those of the authors and do not represent those of The Vanguard Group or its affiliates.

All code and materials are purely for informational and illustrative purposes. They should not be considered investment advice and come with no warranty.

You are using this code at your own risk.

Prerequisites for the Training Workshop

To access the materials in this presentation, please visit:

https://github.com/fintechsteve/modeling-volatility

To successfully complete this training you will need either:

Python with Jupyter Notebooks (e.g. Anaconda)

-or-

An account on Google Colab

Packages used include: pickle, cloudpickle, numpy, pandas, matplotlib, seaborn, scipy and bashtage/arch

Workshop Overview and Timeline

0900 – 0915	**Introductions, Setting up the environment and project background**	
0915 – 0945	Part 1: Working with return data in Python Part 2: Visualizing Timeseries Data	Eunice
0945 – 1000	Part 3: Setting up the Modeling Framework	Steve
1000 – 1030	Part 4a: Understanding Turbulence Signals Part 4b: Creating Turbulence Models	Steve
1030 – 1100	**Break**	
1100 – 1145	Part 5a: Understanding GARCH Estimation Part 5b: Creating GARCH Models	Steve
1145 – 1230	Part 6a: Direct attempts at machine learning	Eunice
1230 – 1300	Conclusion and Q&A	Steve/Eunice

Who is this Presentation For?

Candidate Student #1: Data Scientist with minimal Python experience looking to:

- Better understand how financial modelers think
- Learn traditional financial/econometric modeling techniques for volatility
- Understand the nuances that differentiate data science and trading model development

Candidate Student #2: Finance professional with minimal data science experience looking to:

- Learn how to make the transition from Matlab to Python
- Learn how to leverage financial domain knowledge to better empower data science
- Avoid the pitfalls casual quants make when leveraging more powerful modeling techniques

Setting Up Your Environment

If running on Google Colab, ensure you have a Google account then click on the “Open in Colab” link in the top of each notebook to get started.

Alternatively:

1. Register for Github and clone the project:

   ```
   git clone https://github.com/fintechsteve/modeling-volatility
   ```

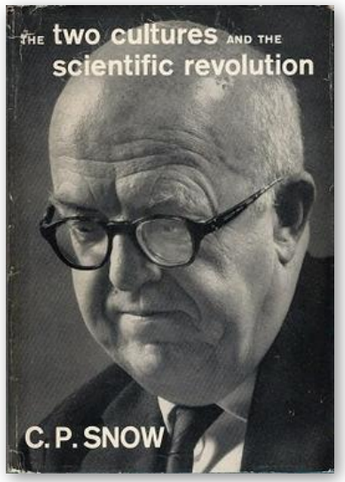
2. Create a new environment (if you don’t have it, now is a good time to get from anaconda.org)

   ```
   conda create -n odsc-volatility python=3.6
   source activate odsc-volatility
   pip install -r requirements.txt
   ```

3. Launch Jupyter Notebooks: `jupyter notebook`

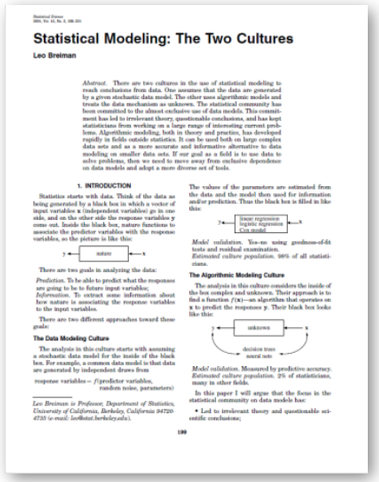
4. If successful, you should be able to run the commands for Part 1 in your browser.

Background (While People Get Their Python Environments Set Up!)



In 1959, **C.P. Snow** wrote an essay titled **“The Two Cultures”** where he observed a general **scientific illiteracy** among otherwise educated individuals.

He **upset many** with his comments – ultimately John Brockman wrote in 1995 about **“The Third Culture”** serving as a mediator.



Statistical Modeling: The Two Cultures

Leo Breiman

Abstract. There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools.

1. INTRODUCTION

Statistics starts with data. Think of the data as being generated by a black box in which a vector of input variables x (independent variables) go in one side, and on the other side the response variables y come out. Inside the black box, nature functions to associate the predictor variables with the response variables, so the picture is like this:

There are two goals in analyzing the data:

Prediction. To be able to predict what the responses are going to be to future input variables;

Information. To extract some information about how nature is associating the response variables to the input variables.

There are two different approaches toward these goals:

The Data Modeling Culture

The analysis in this culture starts with assuming a stochastic data model for the inside of the black box. For example, a common data model is that data are generated by independent draws from

response variables = f(predictor variables, random noise, parameters)

Leo Breiman is Professor, Department of Statistics, University of California, Berkeley, California 94720-4735 (e-mail: leo@stat.berkeley.edu).

The values of the parameters are estimated from the data and the model then used for information and/or prediction. Thus the black box is filled in like this:

Model validation. Yes–no using goodness-of-fit tests and residual examination.

Estimated culture population. 98% of all statisticians.

The Algorithmic Modeling Culture

The analysis in this culture considers the inside of the box complex and unknown. Their approach is to find a function $f(\mathbf{x})$—an algorithm that operates on $\mathbf{x}$ to predict the responses $\mathbf{y}$. Their black box looks like this:

Model validation. Measured by predictive accuracy.

Estimated culture population. 2% of statisticians, many in other fields.

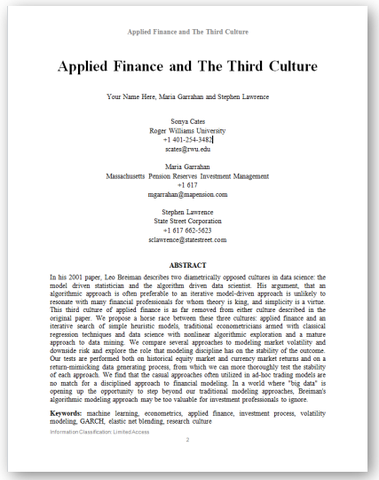
In this paper I will argue that the focus in the statistical community on data models has:

- Led to irrelevant theory and questionable scientific conclusions;

199

In 2001, **Leo Breiman** wrote an essay titled **“Statistical Modeling: The Two Cultures”** where he pitted **traditional statisticians** and their models against **data scientists** and their algorithmic exploration.

Statistical Science published **rebuttal commentary** by D.R. Cox and Brad Efron accompanying Breiman’s article.



Applied Finance and The Third Culture

Applied Finance and The Third Culture

Your Name Here, Maria Garrahan and Stephen Lawrence

Sonya Cates
Roger Williams University
+1 401-254-3483
scates@rwu.edu

Maria Garrahan
Massachusetts Pension Reserves Investment Management
+1 617
mgarrahan@mapension.com

Stephen Lawrence
State Street Corporation
+1 617 662-5623
sclawrence@statestreet.com

ABSTRACT

In his 2001 paper, Leo Breiman describes two diametrically opposed cultures in data science: the model driven statistician and the algorithm driven data scientist. His argument, that an algorithmic approach is often preferable to an iterative model-driven approach is unlikely to resonate with many financial professionals for whom theory is king, and simplicity is a virtue. This third culture of applied finance is as far removed from either culture described in the original paper. We propose a horse race between these three cultures: applied finance and an iterative search of simple heuristic models, traditional econometricians armed with classical regression techniques and data science with nonlinear algorithmic exploration and a mature approach to data mining. We compare several approaches to modeling market volatility and downside risk and explore the role that modeling discipline has on the stability of the outcome. Our tests are performed both on historical equity market and currency market returns and on a return-mimicking data generating process, from which we can more thoroughly test the stability of each approach. We find that the casual approaches often utilized in ad-hoc trading models are no match for a disciplined approach to financial modeling. In a world where “big data” is opening up the opportunity to step beyond our traditional modeling approaches, Breiman’s algorithmic modeling approach may be too valuable for investment professionals to ignore.

Keywords: machine learning, econometrics, applied finance, investment process, volatility modeling, GARCH, elastic net blending, research culture

In 2001, **Leo Breiman** wrote an essay titled **“Statistical Modeling: The Two Cultures”** where he pitted **traditional statisticians** and their models against **data scientists** and their algorithmic exploration.

Working in finance, this “fight” between statisticians and data scientists is **almost** comical!

Professor Sonya Cates (AI professor and former State Street), Maria Garrahan (Investment professional), and Stephen Lawrence co-authored a paper **“Applied Finance and The Third Culture”**

Image Sources: https://en.wikipedia.org/wiki/File:TheTwoCultures.jpg
https://projecteuclid.org/download/pdf_1/euclid.ss/1009213726
Cates, Garrahan and Lawrence, 2018, Applied Finance and the Third Culture

The Intricacy of Simple Financial Models

"Machine learning at its core
is based on inherently simple concepts

Modern finance is built on
decades of intricate modeling"

- Common financial tools including most time series modeling techniques incorporate significant nuance into their design.
- Unless this nuance is captured in inputs into machine learning, it will not compete.
- A compromise is to incorporate financial models and tools as inputs into machine learning models – let ML optimize across models.

Image Source: Office.com. All quotes from Lawrence, Cates and Garrahan (2017), Applied Finance: The Third Culture, Working Paper

Our Currency Volatility Model

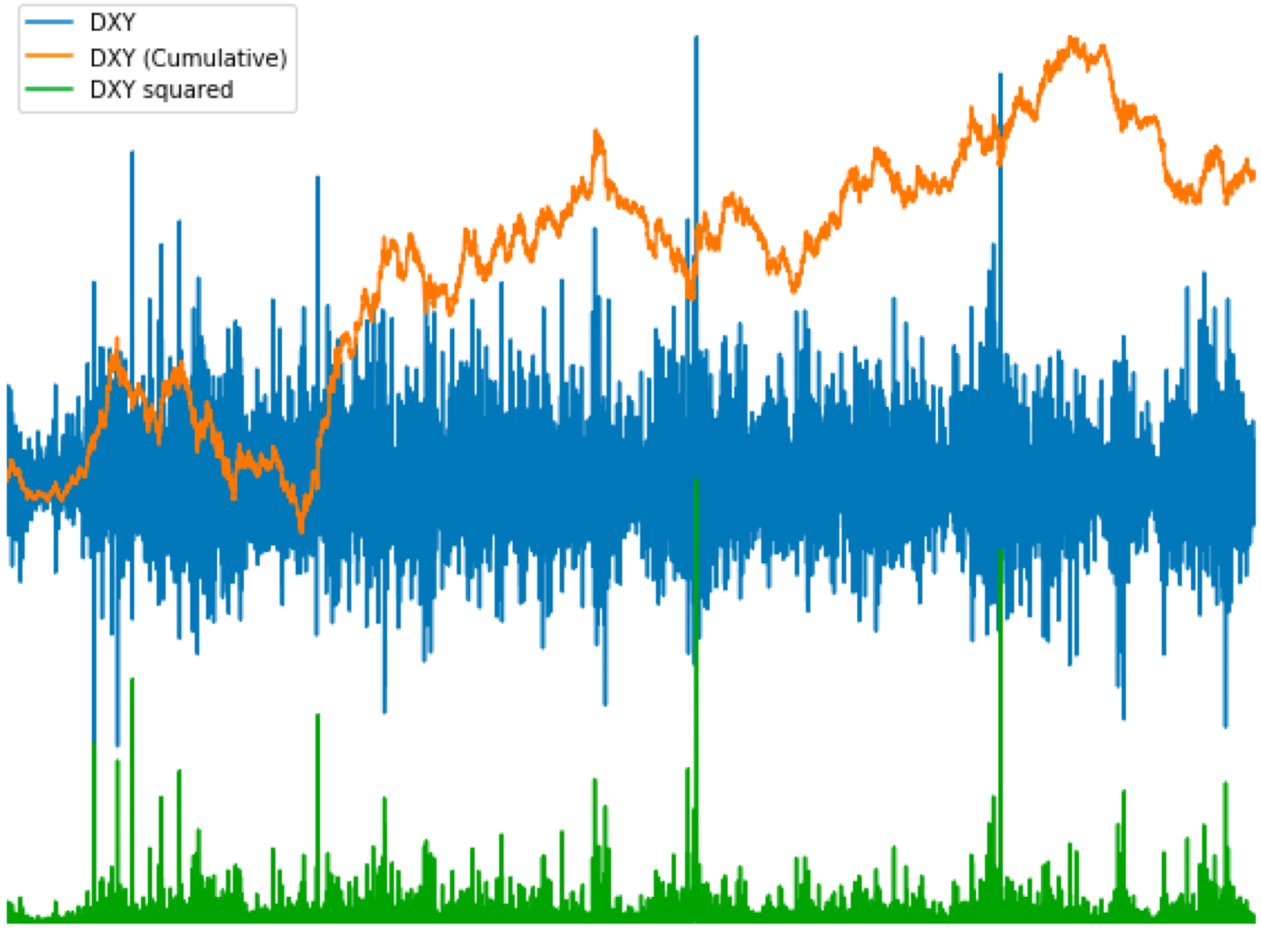
We are interested in predicting tomorrow's currency volatility.

Specifically, we want to predict the volatility of the following 24hr return on a DXY basket

(DXY is a weighted basket of 6 currencies against the dollar)

We have past returns available to us for 9 currencies against the dollar

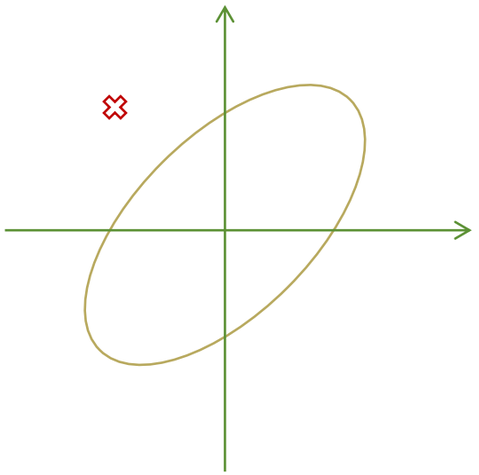
Our goal is to construct a model that assigns a low weight (close to 0) when volatility is likely to be high and a high weight (close to 1) when volatility is likely to be low.



Introduction to Turbulence

Simple “Rule of Thumb”: If the combined returns look abnormal relative to a normal distribution, consider the environment to be “turbulent”

- Mahalanobis Distance Metric: $\mathrm{T} = (x-\mu)'\Sigma^{-1}(x-\mu)$



Assume that abnormalities persist and describe the the overall market volatility (Seems plausible)

When T > 75% of historical T, predict volatility = TRUE

Fine… but how do we parameterize?

- Lookback window for μ and Σ (A year? A quarter? 2 years?)
- Smoothing window for T (Daily? Too fast. Weekly? Monthly? Quarterly?)
- Granularity (Sectors? Industry Groups? Industries?)

Arbitrarily try a few parameters. If they seem to work, go with it. If not, try again.

The GARCH Approach

Assume that all of the returns follow a multivariate GARCH(k) process:

$$\text{Returns follow:} \quad \boldsymbol{r}_t = \boldsymbol{\mu} + \boldsymbol{\Sigma}_t \boldsymbol{\epsilon}_t$$

$$\text{Volatility follows:} \quad \boldsymbol{\Sigma}_t = \boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1 \boldsymbol{\Sigma}_t \boldsymbol{\epsilon}_{t-1} \boldsymbol{\epsilon}'_{t-1} + \cdots + + \boldsymbol{\alpha}_k \boldsymbol{\Sigma}_t \boldsymbol{\epsilon}_{t-k} \boldsymbol{\epsilon}'_{t-k}$$

This is a mathematical model which allows for yesterday's volatility to impact today's volatility.

So our best estimate of whether we are volatile tomorrow can be determine by estimating the parameters in the model using regression and forecasting.

How does a well-behaved econometrician pick k and all the α's?

1. Hold back on 20% of the data to validate stability
2. Use AIC or BIC to penalize complexity. Higher k only allowed if the extra fit outweighs a penalty function for the added parameters this permits.

The Data Science Approach

An algorithmic approach doesn't care about the specific economic model, just observed data

Various variables derived from squared lagged returns likely to be features

e.g. In a boosted tree method, certain combinations of conditions on features are proposed.

Model is optimized over an intelligent search of potential candidate decision rules.

How do you prevent this from becoming a data-mining disaster?

Quick answer: Commit to one run with estimation and holdback data:

1. 20% is held back for the modeler to validate
2. Remaining 80% partitioned into 5x16% chunks
3. Each trial uses 4 chunks to estimate
4. The remaining chunk is used to validate
5. The model with the most stable validation win

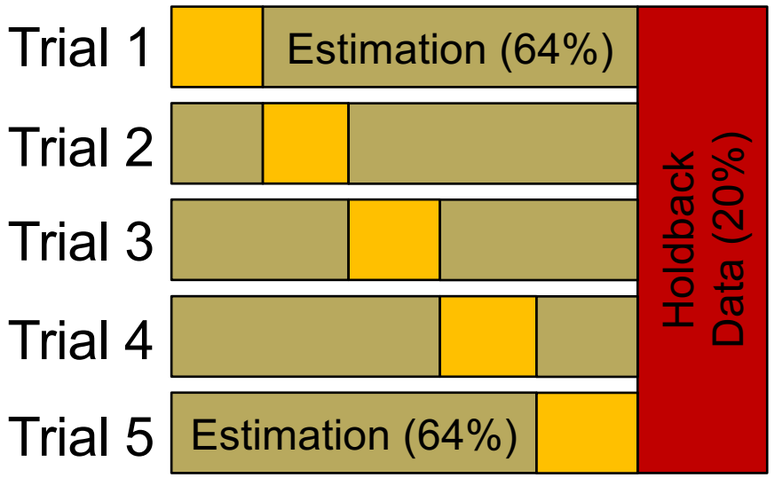


Image source: State Street Global Exchange

Workshop Overview and Timeline

0900 – 0915	Introductions, Setting up the environment and project background	
0915 – 0945	**Part 1: Working with return data in Python** **Part 2: Visualizing Timeseries Data**	**Eunice**
0945 – 1000	Part 3: Setting up the Modeling Framework	Steve
1000 – 1030	Part 4a: Understanding Turbulence Signals Part 4b: Creating Turbulence Models	Steve
1030 – 1100	**Break**	
1100 – 1145	Part 5a: Understanding GARCH Estimation Part 5b: Creating GARCH Models	Steve
1145 – 1230	Part 6a: Direct attempts at machine learning	Eunice
1230 – 1300	Conclusion and Q&A	Steve/Eunice

Workshop Overview and Timeline

0900 – 0915	Introductions, Setting up the environment and project background	
0915 – 0945	Part 1: Working with return data in Python Part 2: Visualizing Timeseries Data	Eunice
0945 – 1000	**Part 3: Setting up the Modeling Framework**	**Steve**
1000 – 1030	Part 4a: Understanding Turbulence Signals Part 4b: Creating Turbulence Models	Steve
1030 – 1100	**Break**	
1100 – 1145	Part 5a: Understanding GARCH Estimation Part 5b: Creating GARCH Models	Steve
1145 – 1230	Part 6a: Direct attempts at machine learning	Eunice
1230 – 1300	Conclusion and Q&A	Steve/Eunice

Important Lessons When Setting Up A Trading Model Backtest

1. Always test your framework with random signals and random returns to help ensure your framework is wrong – Assume you misaligned something and prove to yourself you didn't

2. Start worrying early about the constraints you need your model to have – incorporating them into the data science early will help align your evaluation metrics.

3. Think about ways of reducing the risk of bias in your model.

4. Or appropriately benchmark it to account for the bias.

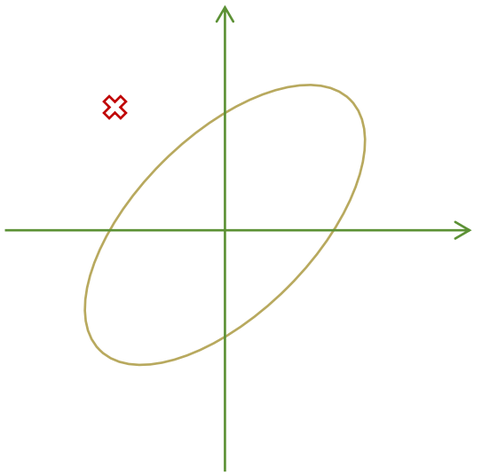
Workshop Overview and Timeline

0900 – 0915	Introductions, Setting up the environment and project background	
0915 – 0945	Part 1: Working with return data in Python Part 2: Visualizing Timeseries Data	Eunice
0945 – 1000	Part 3: Setting up the Modeling Framework	Steve
1000 – 1030	**Part 4a: Understanding Turbulence Signals** **Part 4b: Creating Turbulence Models**	**Steve**
1030 – 1100	**Break**	
1100 – 1145	Part 5a: Understanding GARCH Estimation Part 5b: Creating GARCH Models	Steve
1145 – 1230	Part 6a: Direct attempts at machine learning	Eunice
1230 – 1300	Conclusion and Q&A	Steve/Eunice

Introduction to Turbulence

Simple “Rule of Thumb”: If the combined returns look abnormal relative to a normal distribution, consider the environment to be “turbulent”

- Mahalanobis Distance Metric: $\mathrm{T} = (x - \mu)'\Sigma^{-1}(x - \mu)$



Assume that abnormalities persist and describe the the overall market volatility (Seems plausible)

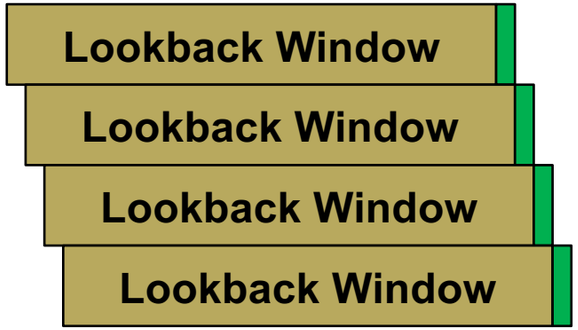
When T > 75% of historical T, predict volatility = TRUE

Fine… but how do we parameterize?

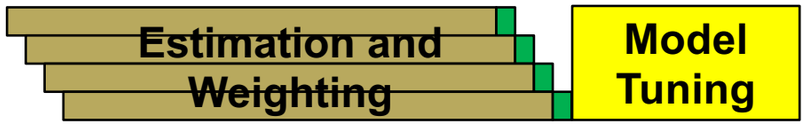
- Lookback window for μ and Σ (A year? A quarter? 2 years?)
- Smoothing window for T (Daily? Too fast. Weekly? Monthly? Quarterly?)
- Granularity (Sectors? Industry Groups? Industries?)

Arbitrarily try a few parameters. If they seem to work, go with it. If not, try again.

Estimation Samples in the Turbulence Model

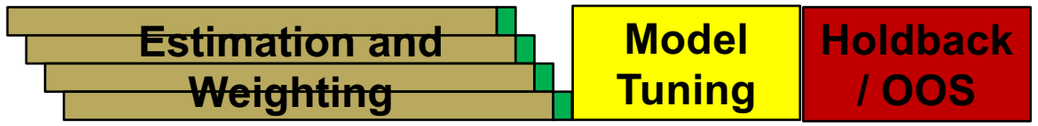


1. Turbulence is estimated with a rolling window



2. Estimates over the period 1/1/1975 – 12/31/1994 are used to determine the optimal weighting scheme

3. 1/1/1995 – 12/31/2004 is used to choose model parameters



4. 1/1/2005 – 12/26/2017 is used as out-of-sample holdback to assess model fit

Important Note

- This workshop explains topics incrementally
- As such, we keep revisiting the same “holdback” period with different models.
- This is not acceptable!
- But it would be even more confusing if we waited until the end for the reveal!

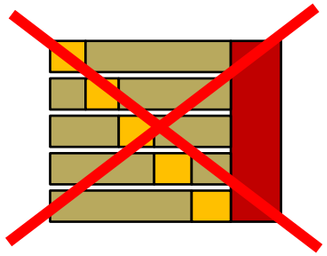
Issues to avoid when creating a model

1. Using future information in today's estimation

Estimation (100%)

2. Using 100% of the sample to estimate model parameters

3. Choosing parameters based solely on in-sample performance



4. Over-reliance on history being the best model of the future and lack of holdback discipline

Workshop Overview and Timeline

0900 – 0915	Introductions, Setting up the environment and project background	
0915 – 0945	Part 1: Working with return data in Python Part 2: Visualizing Timeseries Data	Eunice
0945 – 1000	Part 3: Setting up the Modeling Framework	Steve
1000 – 1030	Part 4a: Understanding Turbulence Signals Part 4b: Creating Turbulence Models	Steve
1030 – 1100	**Break (Start promptly at 1100)**	
1100 – 1145	Part 5a: Understanding GARCH Estimation Part 5b: Creating GARCH Models	Steve
1145 – 1230	Part 6a: Direct attempts at machine learning	Eunice
1230 – 1300	Conclusion and Q&A	Steve/Eunice

Workshop Overview and Timeline

0900 – 0915	Introductions, Setting up the environment and project background	
0915 – 0945	Part 1: Working with return data in Python Part 2: Visualizing Timeseries Data	Eunice
0945 – 1000	Part 3: Setting up the Modeling Framework	Steve
1000 – 1030	Part 4a: Understanding Turbulence Signals Part 4b: Creating Turbulence Models	Steve
1030 – 1100	**Break**	
1100 – 1145	**Part 5a: Understanding GARCH Estimation** **Part 5b: Creating GARCH Models**	**Steve**
1145 – 1230	Part 6a: Direct attempts at machine learning	Eunice
1230 – 1300	Conclusion and Q&A	Steve/Eunice

The GARCH Approach

Assume that all of the returns follow a multivariate GARCH(k) process:

$$\text{Returns follow:} \quad \boldsymbol{r}_t = \boldsymbol{\mu} + \boldsymbol{\Sigma}_t \boldsymbol{\epsilon}_t$$

$$\text{Volatility follows:} \quad \boldsymbol{\Sigma}_t = \boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1 \boldsymbol{\Sigma}_t \boldsymbol{\epsilon}_{t-1} \boldsymbol{\epsilon}'_{t-1} + \cdots + + \boldsymbol{\alpha}_k \boldsymbol{\Sigma}_t \boldsymbol{\epsilon}_{t-k} \boldsymbol{\epsilon}'_{t-k}$$

This is a mathematical model which allows for yesterday's volatility to impact today's volatility.

So our best estimate of whether we are volatile tomorrow can be determine by estimating the parameters in the model using regression and forecasting.

How does a well-behaved econometrician pick k and all the α's?

1. Hold back on 20% of the data to validate stability
2. Use AIC or BIC to penalize complexity. Higher k only allowed if the extra fit outweighs a penalty function for the added parameters this permits.

Workshop Overview and Timeline

0900 – 0915	Introductions, Setting up the environment and project background	
0915 – 0945	Part 1: Working with return data in Python Part 2: Visualizing Timeseries Data	Eunice
0945 – 1000	Part 3: Setting up the Modeling Framework	Steve
1000 – 1030	Part 4a: Understanding Turbulence Signals Part 4b: Creating Turbulence Models	Steve
1030 – 1100	**Break**	
1100 – 1145	Part 5a: Understanding GARCH Estimation Part 5b: Creating GARCH Models	Steve
1145 – 1230	**Part 6a: Direct attempts at machine learning**	**Eunice**
1230 – 1300	Conclusion and Q&A	Steve/Eunice

Workshop Overview and Timeline

0900 – 0915	Introductions, Setting up the environment and project background	
0915 – 0945	Part 1: Working with return data in Python Part 2: Visualizing Timeseries Data	Eunice
0945 – 1000	Part 3: Setting up the Modeling Framework	Steve
1000 – 1030	Part 4a: Understanding Turbulence Signals Part 4b: Creating Turbulence Models	Steve
1030 – 1100	**Break**	
1100 – 1145	Part 5a: Understanding GARCH Estimation Part 5b: Creating GARCH Models	Steve
1145 – 1230	Part 6a: Direct attempts at machine learning	Eunice
1230 – 1300	**Conclusion and Q&A**	**Steve/Eunice**

The Motivation Behind Applied Finance

Finance Relies on Heuristics and Instinct

- Computing power has lagged need, particularly in live visualization.
- Understanding WHY is more important than WHAT (even if loss of accuracy).
- Human reaction time is critical.
- Often there is no time (or budget) for “nerdy” backtesting.

Finance is Often About Telling Stories

The sell side needs to find a simple model that explains recent history.

A buy side modeler needs investors to understand their strategy (no black boxes).

Regulators and auditors need to understand how a risk model works.

Each has an impact on how data science is applied in finance.

Parting Thoughts

Machine Learning and Artificial Intelligence provide powerful tools to derive insight faster and more efficiently than traditional methods.

Different models have varying appeal to differing problems.

The culture of the organization will impact the extent to which a pure data science approach can be tolerated.

With the prevalence of big, unstructured data, a data science approach is going to become more and more necessary in finance.

Understanding your experience path can help guide your approach to data science