

**Certificate Program in Python for Algorithmic Trading**

**Review Questions**

## Weeks 01, 02, & 03

With regard to the topics covered in the first three weeks, you might review the materials based on the following questions. The review questions focus mainly on the **big picture.**

## Python in General

1. Which data types did you learn about?
2. **(Object Type)**

Integer**s}** integer value/ natural number

**Float}** floating point number/ real number

**String**} string object/character, word, text

* + What basic data structures provides Python you with?  
      
    As a general rule, data structures are objects that contain a possibly large number of other objects. Among those that Python provides as built-in structures are:
  + Tuple
  + An immutable collection of arbitrary objects; only a few methods available
  + List
  + A mutable collection of arbitrary objects; many methods available
  + Dict
  + A key-value store object
  + *Set*
  + An unordered collection object for other unique objects

1. What is the basic syntax of a "for loop"?
2. **For** **N in** even**:**
3. **print N**
4. What is the syntax of a list comprehension? (make an example)
5. List comprehensions and functional programming tools like filter (), map (), and reduce () provide means to write code without (explicit) loops that is both compact and in general more readable. lambda or anonymous functions are also powerful tools in this context.  
     
   A specialty of Python is so-called list comprehensions. Instead of looping over existing list objects, this approach generates list objects via loops in a rather compact fashion:

**IN** **[**117)**]:** M = [i\*\*2 for i in range (5)]

M

**OUT [117]: [0,1,4,9,16]**

1. Why is OOP a useful programming paradigm?
2. Used in the right way, it provides a number of advantages compared to, for example, procedural programming. In many cases, OOP seems to be particularly suited for financial modeling and implementing financial algorithms.
3. Natural way of thinking Human thinking typically evolves around real-world or abstract objects, like a car or a financial instrument.
4. OOP is suited to modeling such objects with their characteristics. Reducing complexity via different approaches, OOP helps to reduce the complexity of a problem or algorithm and to model it feature-by-feature. Nicer user interfaces OOP allows in many cases for nicer user interfaces and more compact code.
5. This becomes evident, for example, when looking at the NumPy ndarray class or pandas Data Frame class. Pythonic way of modeling Independent of the pros and cons of OOP, it is simply the dominant paradigm in Python.
6. This is where the saying “everything is an object in Python” comes from. OOP also allows the programmer to build custom classes whose instances behave like every other instance of a standard Python class.
7. This chapter takes a neutral stance, in that OOP is considered an important tool that might not be the best one for every single problem, but that should be at the disposal of programmers and quants working in finance. With OOP, some new language comes along. The most important terms for the purposes of this book and chapter are (more follow later):

With OOP, some new language comes along. The most important terms for the purposes of this book and chapter are (more follow later):

***Class***

An abstract definition of a certain type of objects. For example, a human being.

***Object***

An instance of a class. For example, Sandra.

***Attribute***

A feature of the class (*class attribute*) or of an instance of the class (*instance attribute*). For example, being a mammal, being male or female, or color of the eyes.

***Method***

An operation that the class or an instance of the class can implement. For example, walking.

***Parameters***

Input taken by a method to influence its behavior. For example, three steps.

***Instantiation***

The process of creating a specific object based on an abstract class.

Translated into Python code, a simple class implementing the example of a human being might look as follows:

In [1]: **class** **HumanBeing**(object): [1](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO1-1)

**def** \_\_init\_\_(self, first\_name, eye\_color): [2](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO1-2)

self.first\_name = first\_name [3](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO1-3)

self.eye\_color = eye\_color [4](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO1-4)

self.position = 0 [5](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO1-5)

**def** walk\_steps(self, steps): [6](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO1-6)

self.position += steps [7](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO1-7)

*[1](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#co_object_oriented_programming_CO1-1)*

Class definition statement; self refers to the current instance of the class.

*[2](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#co_object_oriented_programming_CO1-2)*

Special method called during instantiation.

*[3](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#co_object_oriented_programming_CO1-3)*

First name attribute initialized with parameter value.

*[4](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#co_object_oriented_programming_CO1-4)*

Eye color attribute initialized with parameter value.

*[5](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#co_object_oriented_programming_CO1-5)*

Position attribute initialized with 0.

*[6](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#co_object_oriented_programming_CO1-6)*

Method definition for walking with steps as parameter.

*[7](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#co_object_oriented_programming_CO1-7)*

Code that changes the position given the steps value.

Based on the class definition, a new Python object can be instantiated and used:

In [2]: Sandra = HumanBeing('Sandra', 'blue') [1](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO2-1)

In [3]: Sandra.first\_name [2](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO2-2)

Out[3]: 'Sandra'

In [4]: Sandra.position [2](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO2-2)

Out[4]: 0

In [5]: Sandra.walk\_steps(5) [3](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO2-3)

In [6]: Sandra.position [4](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#callout_object_oriented_programming_CO2-4)

Out[6]: 5

*[1](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#co_object_oriented_programming_CO2-1)*

The instantiation.

*[2](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#co_object_oriented_programming_CO2-2)*

Accessing attribute values.

*[3](https://learning.oreilly.com/library/view/python-for-finance/9781492024323/ch06.html#co_object_oriented_programming_CO2-4)*

Calling the method.

## Finance with Python

1. Name three a p p r o a c h e s to price a (European) option in a complete market model.
2. **Black-Scholes is a pricing model:** used to determine the fair price or theoretical value for a call or a put option based on six variables such as volatility, type of option, underlying stock price, time, strike price, and risk-free rate.
3. The model is used to determine the price of a European call option, which simply means that the option can only be exercised on the expiration date

### **The Binomial Option pricing model:** values options using an iterative approach utilizing multiple periods to value American options.

With the model, there are two possible outcomes with each iteration—a move up

or a move down that follow a binomial tree. The model is intuitive and is used

more frequently in practice than the well-known Black-Scholes model.

### **Monte-Carlo Simulation:**

Monte-Carlo simulation is another option pricing model we will consider. The Monte-Carlo simulation is a more sophisticated method to value options. In this method, we simulate the possible future stock prices and then use them to find the discounted expected option payoffs.

### **Risk-neutral Probability**

Before we start discussing different option pricing models, we should understand the concept of risk-neutral probabilities, which are widely used in option pricing and may be encountered in different option pricing models.

**The risk-neutral probability** is a theoretical probability of future outcomes adjusted for risk. There are two main assumptions behind this concept:

1. The current value of an asset is equal to its expected payoff discounted at the risk-free rate.
2. There are no arbitrage opportunities in the market.

The risk-neutral probability is the probability that the stock price would rise in a risk-neutral world. However, we neither assume that all the investors in the market are risk-neutral, nor the fact that risky assets will earn the risk-free rate of return. This theoretical value measures the probability of buying and selling the assets as if there was a single probability for everything in the market.

* + Why is (freedom of) arbitrage such a strong argument in the context of

financial pricing?

* Arbitrage is a type of trade in which a security, currency, or commodity is nearly simultaneously bought and sold, in different markets.
* The purpose of arbitrage is to take advantage of the difference in prices available for the same financial instrument being offered on different exchanges.
* Arbitrage is not only legal in the United States, but is also considered useful to markets as it helps promote market efficiency and also provides liquidity for trading.

In the context of financial pricing (freedom of arbitrage)

Arbitrage provides a mechanism to ensure prices do not deviate substantially from [fair value](https://www.investopedia.com/terms/f/fairvalue.asp) for long periods of time.

With advancements in technology, it has become extremely difficult to profit from pricing errors in the market. Many traders have computerized trading systems set to monitor [fluctuations](https://www.investopedia.com/ask/answers/121714/what-are-differences-between-divergence-and-convergence.asp) in similar [financial instruments](https://www.investopedia.com/terms/f/financialinstrument.asp). Any inefficient pricing setups are usually acted upon quickly, and the opportunity is often eliminated in a matter of seconds. Arbitrage is a necessary force in the financial marketplace.

Arbitrage-free valuation is when price discrepancies are removed, allowing for a more accurate picture of the firm’s valuation based on actual performance metrics.

## How Arbitrage-Free Valuation Works

Arbitrage can severely inflate or deflate the true price of the asset. The firm is doing the same work and has the same underlying capital structure, asset mix, cash flow, and every other metric regardless of what exchange it is listed on or derivative pricing.

Arbitrage-free valuation is used in a couple of different ways. First, it can be the theoretical future price of a security or commodity based on the relationship between spot prices, interest rates [carrying costs](https://www.investopedia.com/terms/c/carrying-costs.asp), exchange rates, transportation costs, [convenience yields](https://www.investopedia.com/terms/c/convenienceyield.asp), etc. Carrying costs are simply the cost of holding inventory

it can also be the theoretical [spot price](https://www.investopedia.com/terms/s/spotprice.asp) of a security or commodity based on the futures price interest rates, carrying costs, convenience yields, exchange rates, transportation costs, etc. Convenience yield is when you hold on to the actual physical good versus the liquid asset. An example would be holding on to a barrel of oil versus holding on to an oil futures contract. When the actual futures price does not equal the theoretical futures price,[arbitrage](https://www.investopedia.com/terms/a/arbitrage.asp) profits may be made.

“”””””””””””””””Arbitrage is more useful for traders rather than investors.”””””””””””””””””

For the trade to be truly risk-free, variables must be known with certainty and transaction costs must be accounted for. Most markets are too efficient to allow risk-free arbitrage trades, hence why there is often times such a large difference in the bid-ask spread. Put another way, the house always wins.

* 1. What is a martingale measure and how can it be used to price options?
  2. Simply stated, in a risk-neutral world, the value of a contingent claim is the discounted expected payoff under the risk-neutral (martingale) measure. This is the probability measure that makes all risk factors (stocks, indices, etc.) drift at the riskless short rate. According to the Fundamental Theorem of Asset Pricing, the existence of such a probability measure is equivalent to the absence of arbitrage.
  3. The Fundamental Theorem of Asset Pricing is one of the cornerstones and success stories of modern financial theory and mathematics.[61]

**The central notion underlying the Fundamental Theorem of Asset Pricing** is the concept of a martingale measure; i.e., a probability measure that removes the drift from a discounted risk factor (stochastic process). In other words, under a martingale measure, all risk factors drift with the risk-free short rate—and not with any other market rate involving some kind of risk premium over the risk-free short rate.

A Simple Example:

*If the market model ℳ is arbitrage-free, then there exists a unique price  associated with any attainable (i.e., replicable) contingent claim (option, derivative, etc.) VT. It satisfies , where e–rT is the relevant risk-neutral discount factor for a constant short rate r.*

* 1. the [risk-free rate of return](https://www.investopedia.com/terms/r/risk-freerate.asp) is one of the most basic components of modern finance. Many of the most famous theories in finance—the [capital asset pricing model (CAPM)](https://www.investopedia.com/terms/c/capm.asp), [modern portfolio theory (MPT)](https://www.investopedia.com/terms/m/modernportfoliotheory.asp) and the [Black-Scholes model](https://www.investopedia.com/terms/b/blackscholes.asp)—use the risk-free rate as the primary component from which other valuations are derived.
  2. The [risk-free asset](https://www.investopedia.com/terms/r/riskfreeasset.asp) only applies in theory, but its actual safety rarely comes into question until events fall far beyond the normal daily [volatile](https://www.investopedia.com/terms/v/volatility.asp) markets. Although it's easy to take shots at theories that use a risk-free asset as their base, there are limited other options.

1. What is the expected utility approach all about and how does it model decision making under uncertainty?

## Understanding Expected Utility

The expected utility of an entity is derived from the expected utility hypothesis. This hypothesis states that under uncertainty, the weighted average of all possible levels of utility will best represent the utility at any given point in time.

Expected utility theory is used as a tool for analyzing situations where individuals must make a decision without knowing which outcomes may result from that decision, i.e., decision making under uncertainty. These individuals will choose the action that will result in the highest expected utility, which is the sum of the products of probability and utility over all possible outcomes. The decision made will also depend on the agent’s risk aversion and the utility of other agents.

This theory also notes that the utility of a money does not necessarily equate to the total value of money. This theory helps explains why people may take out insurance policies to cover themselves for a variety of risks. The expected value from paying for insurance would be to lose out monetarily. But, the possibility of large-scale losses could lead to a serious decline in utility because of diminishing marginal utility of wealth.

* Expected utility refers to the utility of an entity or aggregate economy over a future period of time, given unknowable circumstances.
* It is used to evaluate decision-making under uncertainty.

### **12.4. Valuation**

**A financial asset is a**[**liquid asset**](https://www.investopedia.com/terms/l/liquidasset.asp)**that gets its value from a contractual right or ownership claim. Cash, stocks, bonds, mutual funds, and bank deposits are all are examples of financial assets.**

**One of the most important applications of Monte Carlo simulation is the valuation of contingent claims (options, derivatives, hybrid instruments, etc.).**

**Simply stated, in a risk-neutral world, the value of a contingent claim is the discounted expected payoff under the risk-neutral (martingale) measure.**

**This is the probability measure that makes all risk factors (stocks, indices, etc.) drift at the riskless short rate, making the discounted processes martingales. According to the Fundamental Theorem of Asset Pricing, the existence of such a probability measure is equivalent to the absence of arbitrage.**

**A financial option embodies the right to buy (call option) or sell (put option) a specified financial instrument at a given maturity date (European option), or over a specified period of time (American option), at a given price (strike price). Let us first consider the simpler case of European options in terms of valuation.**

* 1. What problems arise in the context of an incomplete market (model)?

**In this slightly enriched static three state economy the notions of market incompleteness and indeterminacy of the martingale measure are discussed.**

**Super-replication and approximative replication approaches are presented to cope with incompleteness and its consequences for the pricing of contingent claims**

It is well-known by standard results from linear algebra that a basis for the vector space R3R3 needs to consist of three linearly independent vectors. In other words, not every contingent claim is replicable by a portfolio of the traded financial assets. An example is, for instance, the *first Arrow-Debreu security*. The system of linear equations for the replication is

⎨b⋅11+s⋅20=1b⋅11+s⋅10=0b⋅11+s⋅5=0

Therefore, there is no solution to this replication problem.

or those contingent claims that are not attainable C1∈A¯=R3∖AC1∈A¯=R3∖A, the answer is not that simple. Suppose the first Arrow-Debreu security γuγu. It is not replicable as shown above and therefore belongs to the set A¯A¯. Its martingale price is

γu0(q)=11+i⋅EQ((1,0,0)T)=11+i⋅q

* 1. Explain the difference between a static and a dynamic financial model (for pricing purposes).
  2. The difference between a static {*single period}* and a dynamic financial model (for pricing purposes) is that with the static financial modeling for appropriate pricing of financial assets the model fixes the number of possible future states whereas with Dynamic financial modeling there are discrete time dynamics that come into play when pricing financial assets { *Multiperiod models}*

page lof 3

# Tools & Skills

1. Which dual role does the tool conda play?
2. **The open source conda helps with both Python package and virtual environment management.**
3. Why does it make sense to work with Python environments?
4. **Python is the programming language and technology platform of choice, not only for this book but for almost every leading financial institution.**
5. **In combination with a Python 3.7 installation and a secure Jupyter Notebook server installation provides a professional environment for Python development and deployment in the context of Python-for-finance projects.**

**Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python’s simple, easy to** **learn syntax emphasizes readability and therefore reduces the cost of program maintenance**.

**Nowadays, Python is used by the beginner programmer as well as by the highly skilled expert developer, at schools, in universities, at web companies, in large corporations and financial institutions, as well as in any scientific field.**

***Open source***

**Python and the majority of supporting libraries and tools available are open source and generally come with quite flexible and open licenses.**

***Interpreted***

**The reference CPython implementation is an interpreter of the language that translates Python code at runtime to executable byte code.**

***Multiparadigm***

**Python supports different programming and implementation paradigms, such as object orientation and imperative, functional, or procedural programming.**

***Multipurpose***

**Python can be used for rapid, interactive code development as well as for building large applications; it can be used for low-level systems operations as well as for high-level analytics tasks.**

***Cross-platform***

**Python is available for the most important operating systems, such as Windows, Linux, and macOS. It is used to build desktop as well as web applications, and it can be used on the largest clusters and most powerful servers as well as on such small devices as the**[**Raspberry Pi**](http://www.raspberrypi.org/)**.**

***Dynamically typed***

**Types in Python are in general inferred at runtime and not statically declared as in most compiled languages.**

***Indentation aware***

**In contrast to the majority of other programming languages, Python uses indentation for marking code blocks instead of parentheses, brackets, or semicolons.**

***Garbage collecting***

**Python has automated garbage collection, avoiding the need for the programmer to manage memory.**

**The Python Ecosystem A major feature of Python as an ecosystem, compared to just being a programming language, is the availability of a large number of packages and tools.**

**Professional software developers find in Python all they might require to efficiently build large applications. Almost all programming paradigms are supported; there are powerful development tools available; and any task can, in principle, be addressed with Python.**

**There is also another important group of Python users: beginner programmers, i.e., those that are just starting to program. Nowadays, Python has become a very popular language at universities, colleges, and even schools to introduce students to programming.1 A major reason for this is that its basic syntax is easy to learn and easy to understand, even for the non-developer. In addition, it is helpful that Python supports almost all programming styles.2**

1. How do you create an environment with conda?
2. conda create --name $ENVIRONMENT\_NAME
   * How do you install Python packages to such an environment?
   * conda install $PACKAGE\_NAME
3. How do you delete a Python environment?
4. conda deactivate $ENVIRONMENT\_NAME
5. Why is Docker a helpful, platform-independent technology?
6. **Docker containers represent complete filesystems containing all the pieces of a system needed to run certain software, like code, runtime, or system tools. For example, you can run an Ubuntu 18.04 operating system with a Python 3.7 install and the respective Python code in a Docker container hosted on a machine running macOS or Windows 10.**
7. **Docker have taken the IT world by storm. Although the technology is still relatively young, it has established itself as one of the benchmarks for the efficient development and deployment of almost any kind of software application.**
8. **it suffices to think of a Docker container as a separate (“containerized”) filesystem that includes an operating system (e.g., Ubuntu Server 18.04), a (Python) runtime, additional system and development tools, as well as further (Python) libraries and packages as needed. Such a Docker container might run on a local machine with Windows 10 or on a cloud instance with a Linux operating system, for instance.**
9. **The first is a Docker image, which can be compared to a Python class. The second is a Docker container, which can be compared to an instance of the respective Python class.7**
10. **A container is a runtime instance of a Docker image. A Docker container consists of: a Docker image, an execution environment, and a standard set of instructions.**
11. **Docker containers go even further, in that complete filesystems and runtime environments can be easily created in a technically shielded “sandbox” (i.e., the container).**
12. Some reasons to use Docker (for Python Deployment)
13. **Lightweight virtualization/containerization (sharing infrastructure)**
14. **Running different operating systems on a single host system (i.e.: Ubuntu on Mac OS or Windows)**
15. **Quick and easy testing of new technologies (setup, use, dismiss)**
16. **Develop locally (i.e.: notebook), deploy remotely (i.e.: in cloud)**
17. **Reproducibility of set-ups/applications on different infrastructure**
18. **Isolation offset-ups/applications on a host system**
19. How does it help you with managing Python installations?
20. **Docker containers represent complete filesystems containing all the pieces of a system needed to run certain software, like code, runtime, or system tools. For example, you can run an Ubuntu 18.04 operating system with a Python 3.7 install and the respective Python code in a Docker container hosted on a machine running macOS or Windows 10**.
21. What are the benefits of a cloud-based Python installation compared to a local one?

**you have a multitude of benefits when running (production) code in the cloud:**

* **scalability (compute cores, RAM, storage)**
* **availability (24/7, 99.9%, electricity, internet connection)**
* **security (physical access, fire, water, etc.)**

**On the other hand, you wouldn't run a production algo from your local notebook that you carry around with you. Charging problems, shaky internet connectivity, low compute power, etc. are not acceptable in such a context.**

**One benefit of a cloud instance (i.e., a virtual server) compared to a dedicated server rented longer-term is that users generally get charged only for the hours of actual usage; another advantage is that such cloud instances are available literally in a minute or two if needed, which helps with agile development and also with scalability.**

**Although the development and testing of automated algorithmic trading strategies is possible from a local computer (desktop, notebook, etc.), it is not appropriate for the deployment of live strategies trading real money. A simple loss of the web connection or a brief power outage might bring down the whole algorithm, leaving, for example, unintended open positions in the portfolio or causing data set corruption (due to missing out on real-time tick data), potentially leading to wrong signals and unintended trades/positions.**

Why, do you think, a cloud infrastructure is indispensable when running algorithmic trading strategies in automated fashion?

**Deploying Python code for financial applications generally requires high availability, security, and also performance; these requirements can typically only be met by the use of professional compute and storage infrastructure that is nowadays available at attractive conditions in the form of fairly small to really large and powerful cloud instance.**

**From a development and testing point of view, even the smallest Droplet (cloud instance) from Digital Ocean is enough to get started.**

**Operational Risks**: Although the development and testing of automated algorithmic trading strategies is possible from a local computer (desktop, notebook, etc.), it is not appropriate for the deployment of live strategies trading real money. A simple loss of the web connection or a brief power outage might bring down the whole algorithm, leaving, for example, unintended open positions in the portfolio or causing data set corruption (due to missing out on real-time tick data), potentially leading to wrong signals and unintended trades/positions.

Automated Strategy presents a Python script implementing all these aspects and making use of the code from “Online Algorithm”. The script puts the code in a shape that allows, for example, the deployment of the algorithmic trading strategy—based on the persisted algorithm object—on a remote server. It adds both logging and monitoring capabilities based on a custom function that, among others, makes use of ZeroMQ for socket communication. In combination with the short script from “Strategy Monitoring”, this allows for remote real-time monitoring of the activity on a remote server.

# Financial Data Science

0 What are typical real-world problems that you face with regard to (financial) data sets?

* + What are basic approaches to process CSV files with Python?

0 Why is NumPy such a helpful package for numerical computing and financial analytics in particular?

0 Explain the benefits of specialized data structures (eg ndarray object) as

compared to more general ones (eg list object).

What are vectorized operations (with NumPy) and what are the benefits of this programming paradigm?

0 How do you generate random numbers with NumPy?

# Al in Finance {1-10-22}

What do people understand under the Technological Singularity (TS)?

**Or simply the singularity—is a hypothetical point in time at which technological growth becomes uncontrollable and irreversible, resulting in unforeseeable changes to human civilization/ when computers replace human intelligence.**

**The technological singularity is best defined as a point in time when a combination of computer hardware and artificial intelligence algorithms match or exceed the computational ability of the human brain.**

What basic paths consider researchers possible to reach the Financial Singularity?

Artificial Intelligence is beneficial in any domain where patterns have to be found in large quantities of data and effective decisions have to be taken on the basis of those patterns.

The recognition of patterns, traders try to recognize patterns, making decisions based on patterns that they spot. Price formations/price patterns. However, machines are much better at recognizing patterns.

**Emulation**

Maybe one approach to try. The analogy here is we can emulate the complete machine, indistinguishable {not able to identify} Why not apply this approach to the market. We still have human beings still active in the markets, we have machines, trading exchanges, big & small players why not try based on powerful hardware & software to emulate we have all these elements which make up the financial markets. Complete market replication with all agents.

**Algorithms**

May come to the rescue, technological hardware & algorithms performs the calculations. Capable of mimicking what’s going on in the markets. “Every new AI tool getas tested in the crucible (vessel) of finance.”

In order to reach the Financial Singularity, humanity must be capable of producing computer hardware that can match or exceed the computational power of the human brain. Many feel that progress in nanotechnology will pave the way for this outcome.

Computer software is also a limiting factor to the eventuality of the technological singularity. In order to achieve superhuman intelligence as conceived in the definition of the singularity, efficient software capable of modeling and emulating every element of the human brain must be constructed and operate properly.

**Some Discrepant Views of the Singularity**

The possibility of an event like the technological singularity rests on the assumption that all human intelligence is reducible to computing power and that humanity will learn enough about the function of the human mind to “build one” in silicon

Do you personally believe a Technological Singularity is possible and why or why not?

No doubt, we’ll certainly accomplish a lot before 2029 and many aspects of our lives will probably be unrecognizable in 24 years. We’ve already witnessed unprecedented increases in technology, along with advances in genetic engineering, synthetic biology, nootropic drugs, and direct brain-computer interfaces.

I simply don’t see the emergence of a tipping point where we lose ownership of

progress. In my opinion, several factors will keep this from happening:

* As machines get smarter, the remaining challenges that would need to be solved before they can become autonomous will keep getting more complex. The delta between the two will never intersect.
* Societies will form institutions to ensure that AI advancements are constrained by human values. The “mad scientist,” Dr. Frankenstein scenario isn’t realistic. The line between the technology that supports humanity and the technology that threatens it will be policed by smart people who will advise policymakers accordingly.
* As [futurist Martin Ford suggests](http://www.thelightsinthetunnel.com/), well before we get to the point of technological singularity, the technological advancements that lead up to it will displace so many workers in so many skilled professions that society would lose all desire to continue down that path.

What do people understand under the Financial Singularity and how might it be achieved technologically?

* + The singularity – a threshold of time at which Artificial Intelligence’s that are at least smart as humans, and/ or augmented human intelligence, radically remake civilization.

The financial singularity implies that all investment decisions would be better left to a computer program, because the experts with their algorithms have figured out what drives market outcomes and reduced it to a seamless system.

But the futurists’ financial world bears no resemblance to today’s financial world. After all, the financial singularity implies that all prices would be based on such things as optimally projected future corporate profits and the correlation of profits with expected technological innovations and long-term demographic changes. But the smart money hardly ever talks in such ethereal terms.

It seems most likely that AL might pave the way to the financial singularity and an Artificial Financial Intelligence

What is meant by data-driven finance and which technological advances

drive it?

Data -Driven finance is where financial agents become data-processing organisms that can be much better modeled

Science in general has been driven for centuries by the rigorous generation and analysis of data. However, finance used to be characterized by normative theories based on simplified mathematical models of the financial markets, relying on assumptions such as normality of returns and linear relationships. {Expected Utility Theory, Mean Variance Portfolio Theory and the Capital Asset Pricing Theory}

The almost universal and comprehensive availability of (financial) data has led to a shift in focus from a theory-first approach to data-driven finance. Several examples based on real financial data illustrate that many popular financial models and theories cannot survive a confrontation with financial market realities. Although being elegant, they might be too simplistic to capture the complexities, changing nature and non-linearities of financial markets.

This leads to a data-driven instead of a theory- or model-driven view of decision making in financial markets. Financial agents become data-processing organisms that can be much better modeled, for example, by complex neural networks than, say, a simple utility function in combination with an assumed probability distribution.

The technological advances which drives data driven finance are:

complex neural networks, machine learning, subsets, deep learning and reinforcement learning techniques.

T

How does it enable Al-first finance and what do you understand when you hear this expression?

AI-first, theory-free finance might represent a way out of the theory fallacies in traditional finance.

Relying on financial data only and applying general ML and DL algorithms and models to it, are what this book considers AI-first finance

No theories needed, no modeling of human behavior, no assumptions about distributions or the nature of relationships — just data and algorithms. In that sense, AI-first finance could also be labeled theory-free or model-free finance.

* Intelligent algorithms analyze vast amounts of data in the background to uncover anomalies, trends, and exceptions—allowing finance to focus on collaborative actions that blend data insights with financial and operational intuition.
* Finance functions will **harness advancements in technology and data** to help drive effective decision making across the business. Data will be sourced from multiple sources and stored cost-effectively, using cloud technologies to provide the flexibility for rapid scalability.
* Technology – From data lakes to AI and machine learning algorithms, choosing the correct technology to capture, clean, store and analyse data will be key to success.
* Finance functions will harness advancements in technology and data to help drive effective decision making across the business. Data will be sourced from multiple sources and stored cost-effectively, using cloud technologies to provide the flexibility for rapid scalability.

0

What is the difference between a normative and a positive approach to

finance?

The focus is on fundamental concepts from probability theory that build the backbone of quantitative finance.

At its core, financial theory deals with investment, trading and valuation in the presence of uncertainty and risk.

a normative theory is one that is based on assumptions (mathematically axioms) and derives insights, results, and more from the set of relevant assumptions. a positive theory is one that is based on observation, experiment, data, relationships, and the like and describes phenomena given the insights gained from the available information and the derived results

* 1. How does it relate to the history of financial economics and to an Al-first future of finance?

 finance used to be characterized by normative theories based on simplified mathematical models of the financial markets, relying on assumptions such as normality of returns and linear relationships. The almost universal and comprehensive availability of (financial) data has led to a shift in focus from a theory-first approach to data-driven finance. Several examples based on real financial data illustrate that many popular financial models and theories cannot survive a confrontation with financial market realities.

Relying on financial data only and applying general ML and DL algorithms and models to it, are what this book considers AI-first finance. No theories needed, no modeling of human behavior, no assumptions about distributions or the nature of relationships — just data and algorithms. In that sense, AI-first finance could also be labeled theory-free or model-free finance.

1. Name three financial theories that might be considered elegant but not realistic (given empirical support). Not science

#### **Expected Utility, Capital Asset Pricing model, Mean Variance Portfolio theory**

Despite being intellectually appealing, easy-to-implement and mathematically elegant, it is nevertheless surprising that theories and models, such as MVP theory or the CAPM, are still so popular today. First, the popular theories and models presented in this chapter have hardly any meaningful empirical support. Second, some of the theories and models are even theoretically inconsistent with each other in a number of ways. Third, there has been continuous progress on the theoretical and modeling fronts of finance, such that alternative theories and models are available. Fourth, modern computational and empirical finance can rely on almost unlimited data sources and almost unlimited computational power — making concise, parsimonious and elegant mathematical models and results less and less relevant.

page 3of 3