



ADVANCE PERSONAL FINANCE

Trading Algoritmico e Python

Seminario n°1

Ing. Emanuele Colacchi





ALGOCAPITAL.IT



ING.EMANUELE.COLACCHI@GMAIL.COM

Canale
Telegram



Canale
WhatsApp



- Attualmente sono un Trader Istituzionale attivo sui mercati delle commodities.
- Laureato in Ingegneria Gestionale presso la Sapienza Università di Roma.
- Presidente della Commissione Ingegneria Gestionale presso l'Ordine degli Ingegneri della Provincia di Roma.
- Ho maturato una specializzazione in strumenti derivati su indici azionari ed obbligazionari in ottica di medio-lungo periodo.
- Prediligo il trading algoritmico, basato sia su metodi tradizionali che innovativi.
- Docente per FinecoBank SpA, una delle principali banche dirette in Europa
- Coautore di molti libri tra cui «*Python per il Trading*» il primo libro in Italia sul trading algoritmico, «*Value Investing*» e «*Tasse e Investimenti*» pubblicati su Amazon.



LIBRI DI TESTO CONSIGLIATI



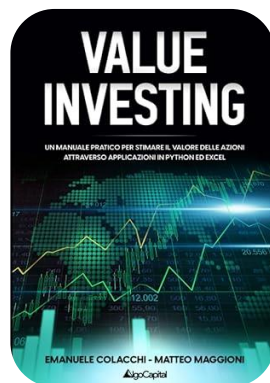
Primo libro in Italia su Python e il Trading che permette di partire da zero per chi vuole imparare a programmare



Approfondimento del volume «Python per il Trading» con moltissimi Trading System programmati e pronti all'uso



Analisi sulla tassazione e strumenti per minimizzare l'impatto dell'imposizione fiscale in Italia

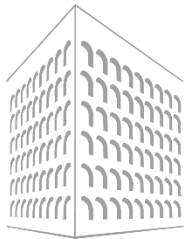


Libro pratico per valutare un titolo attraverso l'Analisi Fondamentale con fogli Excel e Python pronti all'uso.



Tecnica avanzata utilizzata dai principali hedge funds mondiali per identificare coppie di asset correlate

INTRODUZIONE AL TRADING ALGORITMICO



Ordine degli Ingegneri
della Provincia
di Roma

1

**SVILUPPARE UNA STRATEGIA DI
TRADING**

2

**LINEE GUIDA PER LA VALUTAZIONE DEI
RISULTATI**

3

PYTHON PER IL TRADING

Cosa abbiamo visto nel corso base

5

- Caratteristiche dei mercati
- Tipologia di strumenti finanziari (Azioni, Obbligazioni, ETF, Future)
- Analisi Tecnica e Indicatori di Trading
- Candlestick
- Book ed operatività

Cosa non è!

6



GUARDA COME
GUADAGNARE COL TRADING



**4 metodi per fare soldi
online senza investire.**

**Testati e funzionanti al
100%!**



TRUCCHI
PER GUADAGNARE
COL FOREX



**Inizia da Casa tua in soli 10
minuti A partire da Ora!**

Ecco come Guadagno

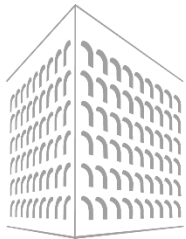


Con le Opzioni Binarie

7



INTRODUZIONE AL TRADING ALGORITMICO



Ordine degli Ingegneri
della Provincia
di Roma

1

SVILUPPARE UNA STRATEGIA DI
TRADING

Componenti Trading System

9



Metodo Generale per sviluppare una strategia vincente

10

Per sviluppare una strategia vincente occorre seguire 4 passi fondamentali:

- **SCEGLI** un numero limitato di indicatori importanti che siano in grado di fornire segnali di trading distinti e non sovrapposti.
- **STRUTTURA** una strategia di trading rispondente alle tue personali inclinazioni.
- **TESTA** il comportamento della strategia nel passato in maniera metodica.
- **TRADA** comportandoti secondo la strategia elaborata con disciplina.

Non esiste né mai esisterà alcuna formula magica dei mercati ma seguendo con serietà una strategia strutturata è possibile ottenere risultati

Cos'è il Trading Meccanico?

- E' una modalità di trading basata su un insieme di regole che definiscono completamente la vita di un trade.
- Entry, exit, stop loss, position sizing sono chiaramente predefinite.
- Un sistema meccanico non lascia spazio a decisioni discrezionali.
- Le regole sono sviluppate tramite una ricerca rigorosa e attraverso test su dati storici.
- Ogni trade ha un esito casuale.
- Backtesting estensivi possono dimostrare un ritorno positivo sul lungo termine.

Benefici del Trading Meccanico

- Mantiene a distanza le emozioni.
- Non c'è da discutere sul segnale.
- I trading system possono essere valutati sulle probabilità.
- Le performance attese (profitti, drawdowns, win/loss ratio) sono conosciute.

Benefici del Trading Meccanico

- Alcuni dei migliori trade sono quelli in cui le personali emozioni direbbero di non entrare
- Permette di andare contro il sentire comune.
- È simile a stare in un casino ma dalla parte del proprietario e non del giocatore perchè sa come portare le probabilità a suo favore.
- Porta un po' di certezza in un ambiente altamente incerto.

Comprare o Progettare? - Comprare

- É l'unica opzione percorribile se si ha mancanza di tempo o mancano abilità di programmatore.
- Prestare molta attenzione a sistemi “black box” che danno solo indicazione “luce verde/luce rossa”.
- Se non si conosce bene come il sistema di trading funziona si potrebbe non credere completamente al sistema e quindi esitare nell'eseguire il segnale. Le emozioni in questo caso riprenderebbero il sopravvento.

Comprare o Progettare? - Progettare

- Richiede forti competenze di programmazione e MOLTO tempo.
- Se si progetta il sistema, lo si costruisce e lo si testa si avrà completa fiducia in esso e più difficilmente si esiterà nell'eseguire il segnale.
- Nessun trading system funziona per sempre. Se si progetta potrebbe essere più facile settarlo, mantenendolo sempre aggiornato.

Strumenti necessari

- Software per Grafici e Backtesting
 - Trading View
 - Tradesignal
 - Bloomberg
 - Tradestation
 - ProRealTime
- Dati storici affidabili
- I dati sui future devo subire un processo di “backadjust”.

Domande preliminari

- Cosa si vuole dal proprio sistema?
- Quanto tempo è possibile dedicare a fare trading con il sistema? Ho già un lavoro a tempo pieno?
- Time frame da utilizzare. Grafico intraday, giornaliero, settimanale ?
- Quale mercato tradare? Azionario, forex, commodities, ecc?

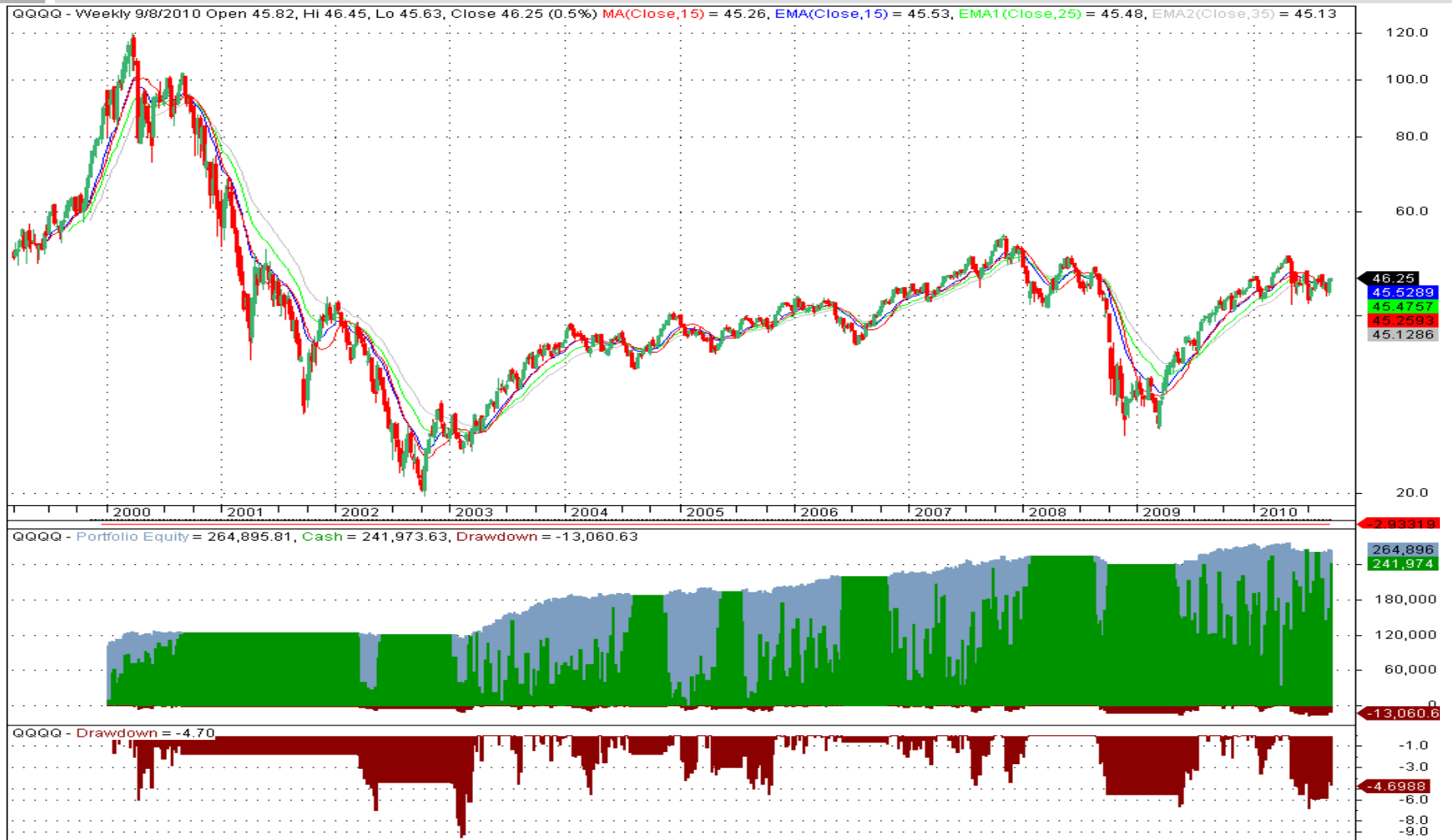
Backtesting e Ottimizzazione

- Passi: Scegliere un concetto, programmarlo, valutare i risultati, modificare i concetti di partenza, ripetere se necessario.
- Le idee nascono dalla lettura dei libri, trading magazines, online blogs, ecc.
- Almeno devono esserci 100 trades per essere statisticamente significativi. Maggiore è il numero trades, maggiore sarà l'affidabilità del sistema.
- Verificare che i profitti non derivino solo da pochi trades.

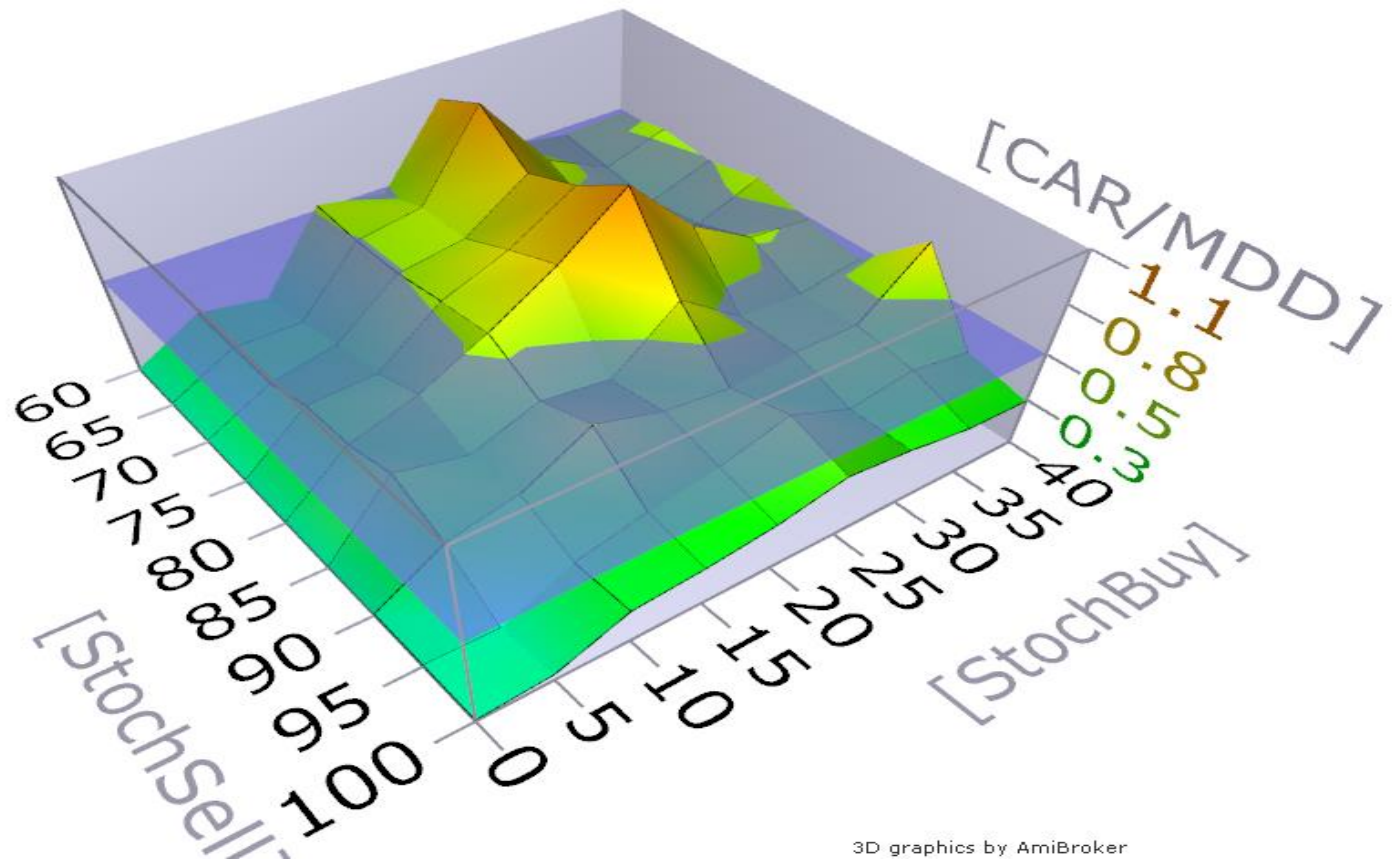
Backtesting e Ottimizzazione

- Osservare l'Equity Line. Da preferire una dolce pendenza crescente.
- Osservare il Drawdown.
- Ribaltare il segnale per conferma o per rendere un sistema non profittevole in uno profittevole.
- Essere cauti se trovare una serie di regole che funziona bene, ha buoni risultati ma manca di senso logico.
- Il sistema sarà più solido se funziona su più mercati.

Equity & Drawdown



Ottimizzazione 3D

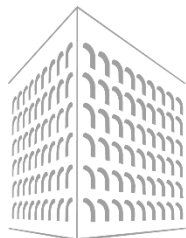


3D graphics by AmiBroker

Tipologie di Trading System

- Trend Following
- Mean Reversion
- Arbitraggi statistici

INTRODUZIONE AL TRADING ALGORITMICO



Ordine degli Ingegneri
della Provincia
di Roma

2

LINEE GUIDA PER LA VALUTAZIONE DEI RISULTATI

Variabili

24

Gross Profit: è la somma di tutti i profitti.

Gross Loss: è la somma di tutte le perdite.

Total Net Profit: è la differenza tra Gross Profit e Gross Loss. Se la differenza è positiva allora il *trading system* ha generato un profitto, in caso contrario una perdita.

Profit Factor: è il rapporto in valore assoluto tra Gross Profit e Gross Loss. Questo rapporto esprime il rischio associato ad un determinato ammontare di profitto. Deve essere almeno maggiore di 1. È da preferire un valore elevato poiché significa che ad un aumento del rischio corrisponde un aumento più che proporzionale del profitto.

Variabili

25

Number of Winning Trades: è il numero di trade a profitto.

Number of Losing Trades: è il numero di trade in perdita.

Total Number of Trades: è il numero totale di trade.

Percent Profitable: è il rapporto tra numero di trade a profitto ed il numero totale di trade. Non rappresenta però una voce determinante nella valutazione dell'affidabilità di un *trading system*.

Variabili

26

Average Winning Trade: è la media del profitto generato dai trade positivi.

Average Losing Trade: è la media della perdita generata dai trade negativi.

Ratio AvgWin/AvgLos: è il rapporto tra Average Winning Trade e Average Losing Trade. È da preferire un valore elevato poiché significa che ad un aumento del rischio medio corrisponde un aumento più che proporzionale del profitto medio.

Average Bars in Winning Trades: è la durata media in termini di tempo dei trade a profitto.

Average Bars in Losing Bars: è la media in termini di tempo dei trade in perdita.

Variabili

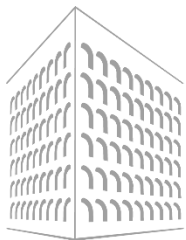
27

Max Drawdown: è il massimo valore di declino dell'Equity Line (linea dei profitti).

Return on Account: è la percentuale di valorizzazione del capitale investito.

Percent of Time in the Market: è la percentuale di tempo con posizioni aperte. Un valore elevato di questa percentuale corrisponde ad un maggior rischio.

INTRODUZIONE AL TRADING ALGORITMICO



Ordine degli Ingegneri
della Provincia
di Roma

3

PYTHON

Bibliografia



Le Basi di
Python

Grafici Tecnici

Serie Storiche

Simulazioni
Montecarlo

Gestione di
portafoglio

Generare
Report

Disponibile su
amazon

Analisi Mercato
Azionario

Gestire la
Volatilità

1 # Cos'è Python

Evoluzione dei mercati finanziari

ALGOCAPITAL.IT



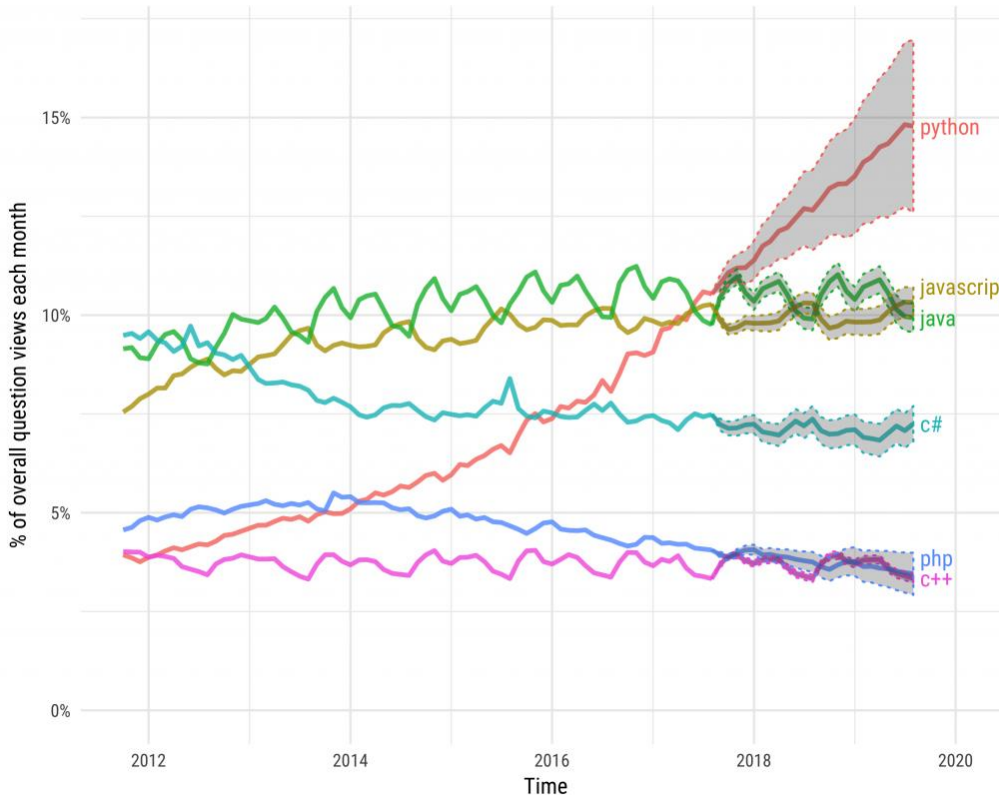
8 Febbraio 1971: avvio negoziazioni sul mercato NASDAQ.

Da mercato alle grida...

...A mercato elettronico

1 # Cos'è Python

Evoluzione dei linguaggi di programmazione



ALGOCAPITAL.IT

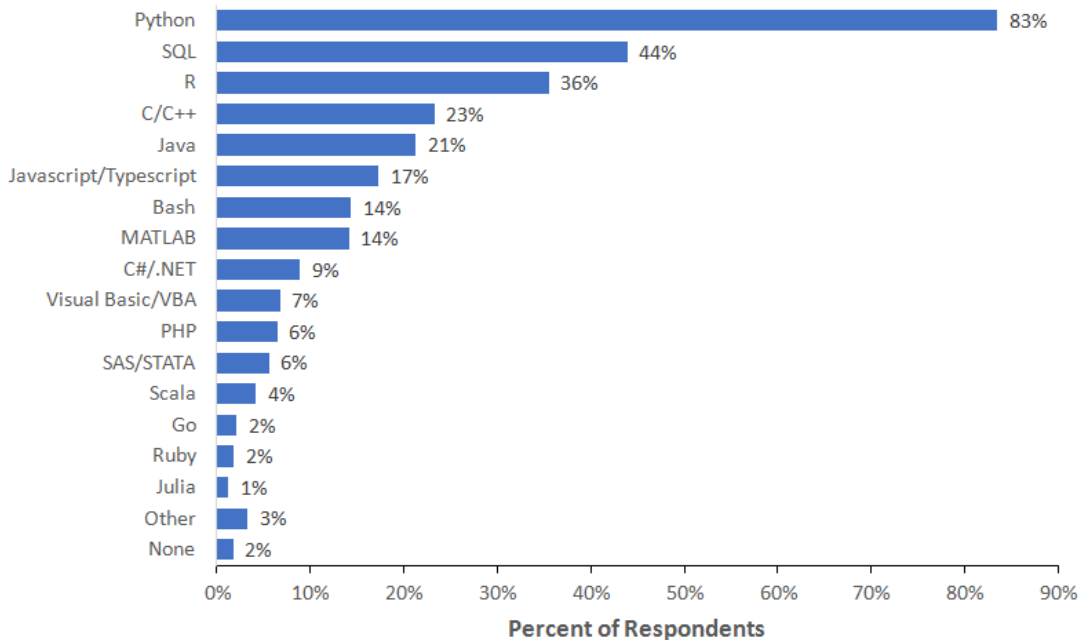
- Python è l'unico linguaggio che sta vivendo una fase di crescita inarrestabile.
- Questo avviene perché viene utilizzato in moltissimi progetti di machine learning, IoT, data science e finanza.
- Tali settori stanno rivoluzionando il mondo sia per gli utenti business che per il mondo consumer, ed è normale che gli sviluppatori abbiano iniziato ad approfondire il linguaggio di programmazione che anima gran parte dei progetti più attuali.

1 # Cos'è Python

Evoluzione dei linguaggi di programmazione

ALGOCAPITAL.IT

What programming language do you use on a regular basis?



I punti di forza principali sono:

- Flessibilità ed adattabilità a molteplici scopi/progetti.
- Community dinamica di utilizzatori in forte crescita.
- Alta velocità di apprendimento.
- Ha sostituito Java in tutte le università.
- Multiplatforma.
- Numerose librerie disponibili.

.....

- **Gratuito !**

Python è un linguaggio di programmazione ad alto livello, rilasciato pubblicamente per la prima volta nel 1991 dal suo creatore Guido van Rossum.

Deriva il suo nome dalla commedia **Monty Python's Flying Circus** dei celebri Monty Python, in onda sulla BBC nel corso degli anni 70.

Attualmente, lo sviluppo di Python viene gestito dall'organizzazione no-profit **Python Software Foundation**.

1 # Cos'è Python

Lo Zen di Python (1)

ALGOCAPITAL.IT

- Bello è meglio di brutto.
- Esplicito è meglio di implicito.
- Semplice è meglio di complesso.
- Complesso è meglio di complicato.
- Lineare è meglio di nidificato.
- Rado è meglio di denso.
- La leggibilità è importante.
- I casi speciali non sono abbastanza speciali per infrangere le regole.
- Anche se la praticità batte la purezza.

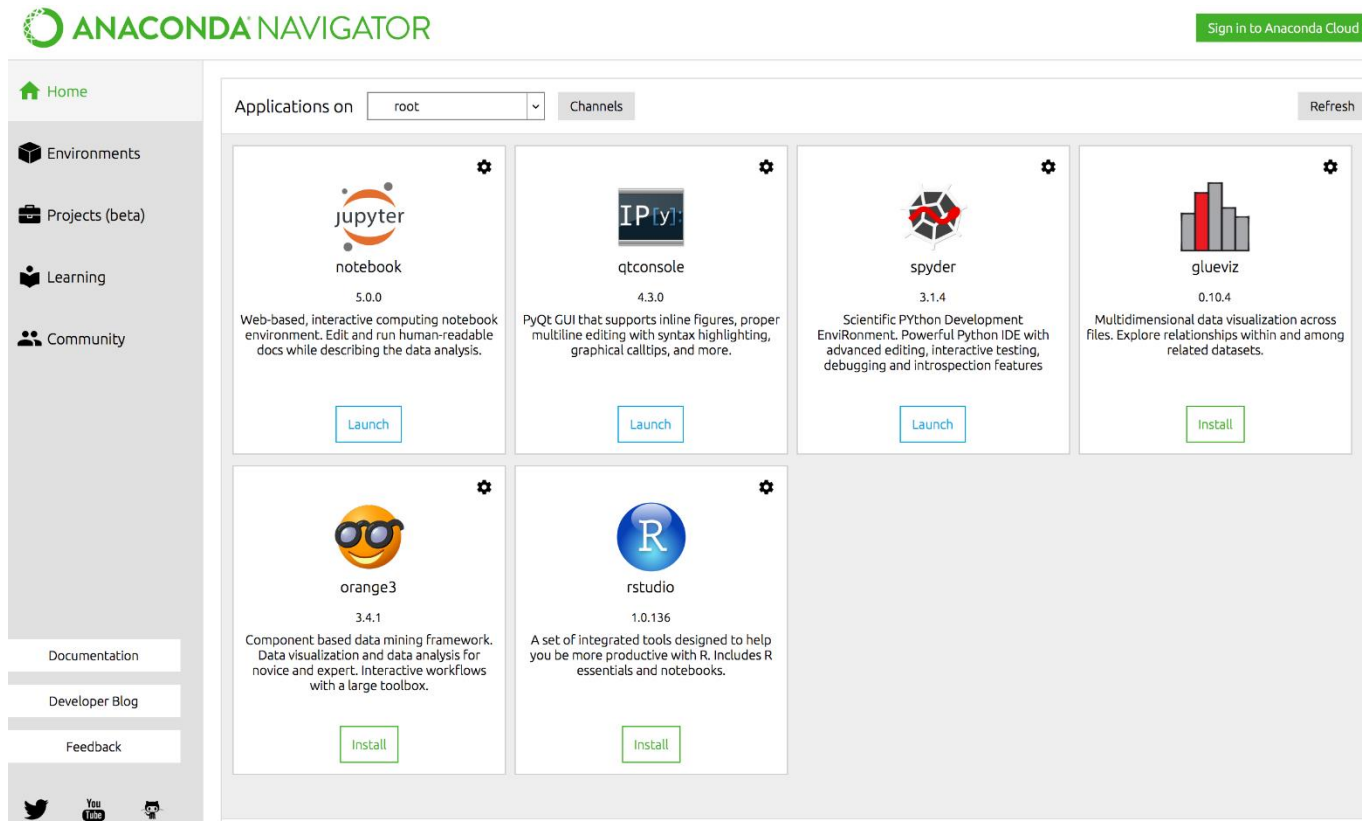
Lo Zen di Python (2)

- Gli errori non dovrebbero mai accadere in silenzio.
- Salvo esplicitamente silenziati.
- Davanti all'ambiguità, rifiuta la tentazione di indovinare.
- Ci dovrebbe essere un - e preferibilmente un solo - modo ovvio per farlo.
- Anche se all'inizio potrebbe non essere un modo ovvio, a meno che tu non sia olandese.
- Adesso è meglio che mai.
- Anche se "mai" è spesso meglio di "proprio adesso".
- Se l'implementazione è difficile da spiegare, è una cattiva idea.
- Se l'implementazione è facile da spiegare, potrebbe essere una buona idea.
- I namespace sono una delle grandi idee - facciamone di più!

Esistono innumerevoli possibilità di dotarsi di Python, più o meno immediate o personalizzabili.

Una soluzione ottimale è procedere all'installazione utilizzando Anaconda, (www.anaconda.com) il quale fornisce una procedura standard, sicura ed affidabile.





STRING

LIST / SET

IF

FUNCTION

INT / FLOAT

TUPLE

FOR

CLASS

BOOLEAN

DICTIONARY

WHILE

LIBRARY

Le stringhe sono sequenze di caratteri, in codifica Unicode UTF-8.

Nel codice Python le stringhe sono rappresentate come una sequenza di caratteri fra apici. Possono essere utilizzati indifferentemente apici singoli, oppure il doppio apice e sono composte da unità più piccole: i caratteri.

```
word = "Hello World"
```

CODICE

```
>>> print word
```

```
Hello World
```

1 # Cos'è Python

String

ALGOCAPITAL.IT

"hello"+"world"	"helloworld"	# concatenation
"hello"*3	"hellohellohello"	# repetition
"hello"[0]	"h"	# indexing
"hello"[-1]	"o"	# (from end)
"hello"[1:4]	"ell"	# slicing
len("hello")	5	# size
"hello" == "Hello"	False	# comparison
"e" in "hello"	1	# search

Python supporta numeri interi e decimali definiti dalla presenza o meno della virgola.

Se in operazioni matematiche anche un solo numero è float il risultato implicitamente diviene un float.

```
>>> 1 + 2.0
3.0
>>> int(2.3)
2
>>> float(5)
5.0
```

CODICE

I valori booleani rappresentano due valori fondamentali in programmazione: **True** e **False**.

Possono verificare il valore di un'espressione, confrontare variabili, decidere quando interrompere un'iterazione o un intero programma.

Operatore	Significato	Esempio
or	or logico	<code>x or y</code>
and	and logico	<code>x and y</code>
not	negazione	<code>not x</code>

Operatore	Funzione	Esempi
>	maggiore	$a > b$
<	minore	$a < b$
<=	minore od uguale	$a \leq b$
>=	maggiore od uguale	$a \geq b$
==	uguale	$a == b$
!=	diverso	$b != b$

Esistono quattro tipi di collezioni di dati:

- **Lista:** è una raccolta ordinata e modificabile. Consente membri duplicati.
- **Tupla:** è una raccolta ordinata e immutabile. Consente membri duplicati.
- **Set:** è una raccolta non ordinata e non indicizzata. Nessun duplicato.
- **Dizionario:** è una raccolta non ordinata, modificabile e indicizzata. Nessun duplicato.

Una **lista** è una serie ordinata di valori, ognuno identificato da un indice.

I valori che fanno parte della lista sono chiamati elementi. Le liste sono simili alle stringhe essendo insiemi ordinati di caratteri, fatta eccezione per il fatto che gli elementi di una lista possono essere di tipo qualsiasi.

```
>>> this_list = ["ISP.IM", "UCG.IM", "LDO.IM"]  
>>> print(this_list[0])
```

CODICE

```
ISP.IM
```

Le **tuple** sono sequenze simili alle liste, ma, a differenza delle liste, sono sequenze immutabili. Contengono oggetti eterogenei, identificati da un indice numerico.

Le tuple sono rappresentate come insiemi di valori separati da virgole e racchiusi fra parentesi tonde; per il resto abbiamo operatori e notazioni analoghe a quelle delle liste.

```
>>> this_tuple = ("ISP.IM", "UCG.IM", "LDO.IM")  
>>> print(this_tuple[1])
```

CODICE

```
UCG.IM
```

I dizionari sono vettori associativi, quello che in altri linguaggi di programmazione viene anche indicato con "hash" o "mappings". Somigliano a liste, ma gli indici (**keys** o chiavi) sono stringhe, od altri oggetti Python immutabili, ad esempio **tuple**.

Si chiamano dizionari perché l'uso di stringhe per ritrovare elementi di un insieme è analogo a quello che si fa quando si cerca una parola in un dizionario. Si usano per trattare insiemi di elementi contraddistinti da un nome, piuttosto che da un indice numerico, come per le liste.

I dizionari sono oggetti mutabili ed un dizionario può contenere oggetti eterogenei: numeri, stringhe, liste, dizionari. Si possono costruire strutture complesse con dizionari e liste annidate.

```
this_dict = {  
    "ticker": "UCG.IM",  
    "name": "UniCredit",  
    "price": 12.36  
}  
  
# per accedere ai valori  
  
x = this_dict["name"]  
print(x)
```

CODICE

If è l'istruzione condizionale più semplice che permette di far seguire differenti percorsi al flusso di esecuzione di un programma a seconda del risultato della valutazione.

```
if x > 0:  
    print ("x è positivo")
```

CODICE

Fondamentale in Python è **l'indentazione**, altrimenti viene visualizzato un messaggio di errore.

1 # Cos'è Python

IF – ELIF - ELSE

ALGOCAPITAL.IT

CODICE

```
a = 200
b = 33

if b > a:
    print("b maggiore di a")

elif a == b:
    print("a e b sono uguali")

else:
    print("a è maggiore di b")
```

Il ciclo for ci permette di iterare su tutti gli elementi di un iterabile ed eseguire un determinato blocco di codice. Un iterabile è un qualsiasi oggetto in grado di restituire tutti gli elementi uno dopo l'altro, come ad esempio liste, tuple, set, dizionari (restituiscono le chiavi), ecc.

```
azioni = ["ISP.IM", "UCG.IM", "LDO.IM"]
```

CODICE

```
for x in azioni:  
    print(x)
```

Il ciclo **while** itera fintanto che una condizione è vera:

```
i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i += 1
```

CODICE

Le funzioni sono uno strumento che ci permette di raggruppare un insieme di istruzioni che eseguono un compito specifico. Le funzioni accettano in input zero o più argomenti (o parametri), li elaborano, e restituiscono in output un risultato.

Una volta definita una funzione, è possibile eseguirla, passando argomenti diversi a seconda della situazione.

Questo ci permette di rendere il codice più ordinato ed evitare ripetizioni.

```
>>> def quadrato(n):
```

```
>>>     return n**2
```

```
>>> x = quadrato(5)
```

```
25
```

```
>>> quadrato(quadrato(5))
```

```
625
```

CODICE

Python è un linguaggio orientato ad oggetti. La classe più semplice che possiamo definire in Python è:

```
>>> class Test:  
>>>     pass
```

CODICE

In una classe abbiamo metodi che sono simili alle funzioni con due differenze:

- I metodi sono definiti all'interno della definizione di classe per rendere più esplicita la relazione tra la classe ed i metodi corrispondenti.
- La sintassi per invocare un metodo è diversa da quella usata per chiamare una funzione.

1 # Cos'è Python

Le librerie: Numpy

ALGOCAPITAL.IT

NumPy è una libreria open source per il linguaggio di programmazione Python, che aggiunge supporto a grandi matrici e array multidimensionali insieme a una vasta collezione di funzioni matematiche di alto livello per poter operare efficientemente su queste strutture dati.

È stato creato nel 2005 da Travis Oliphant basandosi su Numeric di Jim Hugunin.

<https://numpy.org/>

```
import numpy as np
```

CODICE

```
>>> a = np.array([2,3,4])
```

```
>>> a
```

```
array([2, 3, 4])
```

```
>>> a.dtype
```

```
dtype('int64')
```

```
>>> b = np.array([1.2, 3.5, 5.1])
```

```
>>> b.dtype
```

```
dtype('float64')
```

Pandas è una libreria software scritta per la manipolazione e l'analisi dei dati.

In particolare, offre strutture dati e operazioni per manipolare tabelle numeriche e serie temporali.

Il nome deriva dal termine "panel data", termine econometrico per set di dati che include osservazioni su più periodi di tempo per gli stessi individui.

<https://pandas.pydata.org/>

Pandas si interfaccia in maniera flessibile con una molteplicità di diversi altri tipi di file:

- CSV
- JSON
- HTML
- Excel
- SQL
- ...

Inoltre ha ottimi metodi di compressione per lavorare su Big Data.

1 # Cos'è Python

Le librerie: Pandas

ALGOCAPITAL.IT

	A	B	C	D	E	F
1		COUNTRY	POP	AREA	GDP	CONT
2	CHN	China	1398.72	9596.96	12234.78	Asia
3	IND	India	1351.16	3287.26	2575.67	Asia
4	USA	US	329.74	9833.52	19485.39	N.America
5	IDN	Indonesia	268.07	1910.93	1015.54	Asia
6	BRA	Brazil	210.32	8515.77	2055.51	S.America
7	PAK	Pakistan	205.71	881.91	302.14	Asia
8	NGA	Nigeria	200.96	923.77	375.77	Africa
9	BGD	Bangladesh	167.09	147.57	245.63	Asia
10	RUS	Russia	146.79	17098.25	1530.75	
11	MEX	Mexico	126.58	1964.38	1158.23	N.America

```
>>> df = pd.read_excel('data.xlsx', index_col=0)
>>> df
```

	COUNTRY	POP	AREA	GDP	CONT
CHN	China	1398.72	9596.96	12234.78	Asia
IND	India	1351.16	3287.26	2575.67	Asia
USA	US	329.74	9833.52	19485.39	N.America
IDN	Indonesia	268.07	1910.93	1015.54	Asia
BRA	Brazil	210.32	8515.77	2055.51	S.America
PAK	Pakistan	205.71	881.91	302.14	Asia
NGA	Nigeria	200.96	923.77	375.77	Africa
BGD	Bangladesh	167.09	147.57	245.63	Asia
RUS	Russia	146.79	17098.25	1530.75	NaN
MEX	Mexico	126.58	1964.38	1158.23	N.America

Matplotlib è una libreria per la creazione di grafici per il linguaggio di programmazione Python e la libreria matematica NumPy.

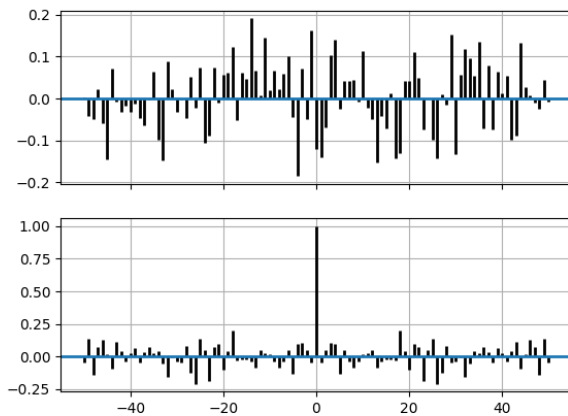
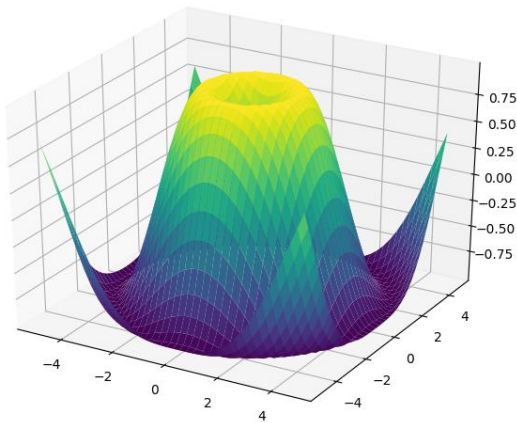
Fornisce API orientate agli oggetti che permettono di inserire grafici all'interno di applicativi usando toolkit GUI generici, come WxPython, Qt o GTK.

È progettata per assomigliare a MATLAB.

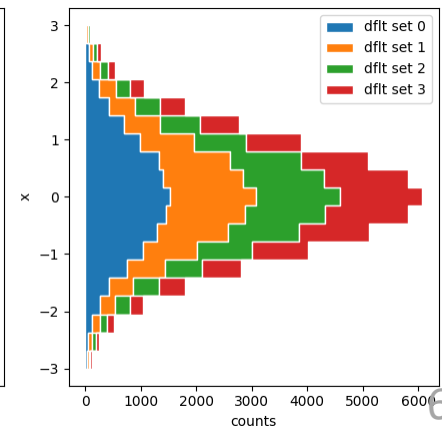
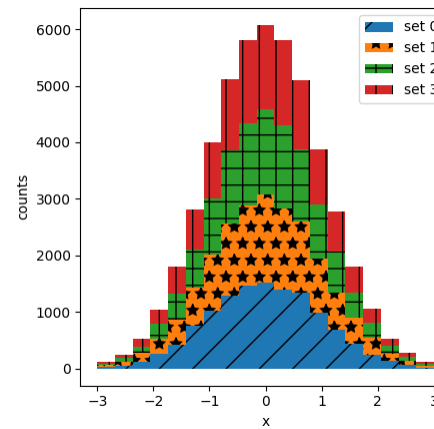
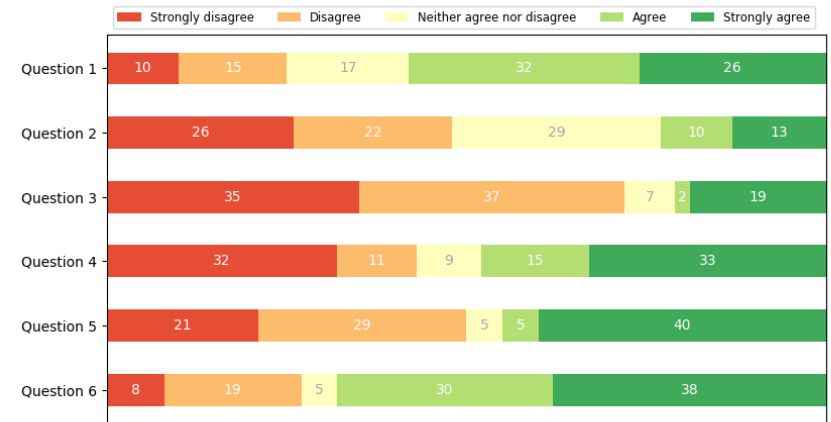
<https://matplotlib.org/>

1 # Cos'è Python

Le librerie: Matplotlib



ALGOCAPITAL.IT



- Yahoo Finance (<https://finance.yahoo.com>)
- FMP (<https://site.financialmodelingprep.com/>)
- Bloomberg (<https://www.bloomberg.com>)
- Reuters (<https://www.reuters.com>)
- Alpha Vantage (<https://www.alphavantage.co>)
- Google Finance (<https://www.google.com/finance>)
- IEX (<https://iextrading.com>)
- World Trading Data (<https://www.worldtradingdata.com>)
- UniBit (World Trading Data (<https://unibit.ai/pricing>))



pip install yfinance

CODICE

```
import pandas as pd
import yfinance as yf
import datetime
```

```
# Selezione data iniziale e data finale
```

```
start = datetime.datetime(2017, 1, 1)
end = datetime.datetime(2023, 3, 1)
```

```
# Salvataggio dati dell'azione Intesa San Paolo
```

```
data = yf.download('ISP.MI', start, end)
```

Per esportare i dati salvati in un Pandas Dataframe è sufficiente una singola riga di codice.

L'integrazione con Excel è quindi immediata e flessibile.

```
data.to_excel('ISP.xlsx')
```

CODICE



Graficare i dati e costruire indicatori

Costruzione SMA a 50 periodi

CODICE

```
data['50D SMA'] = data['Adj Close'].rolling(50).mean()
```

Costruzione SMA a 200 periodi

```
data['200D SMA'] = data['Adj Close'].rolling(200).mean()
```

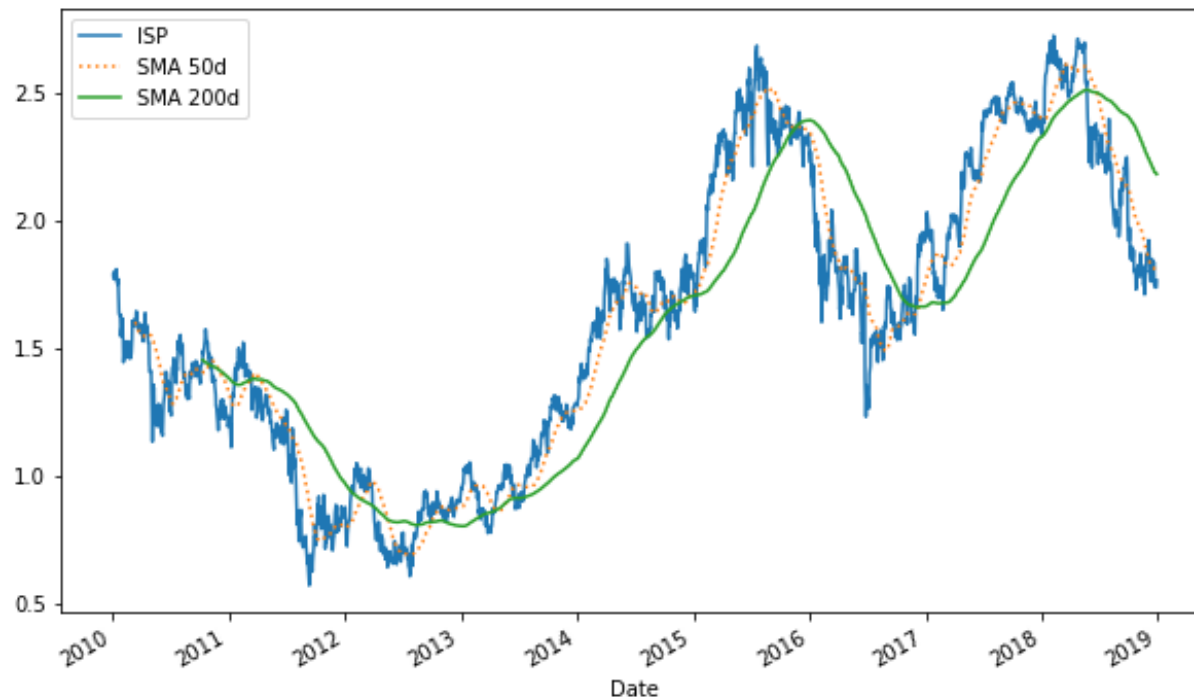
Costruzione del grafico

```
data['Adj Close'].plot()
```

```
data['50D SMA'].plot()
```

```
data['200D SMA'].plot()
```

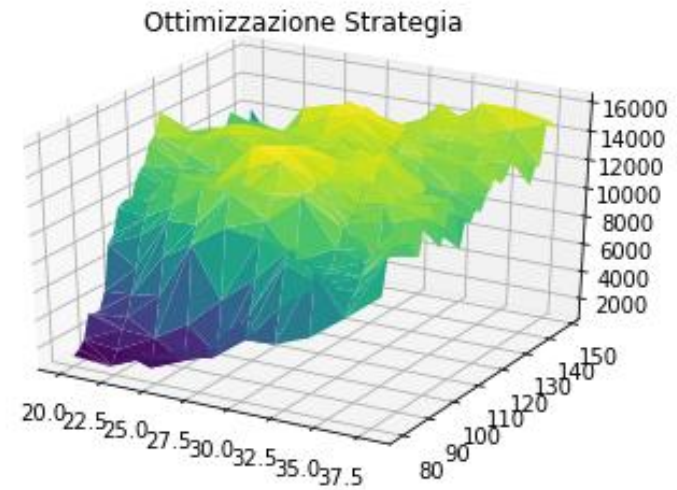
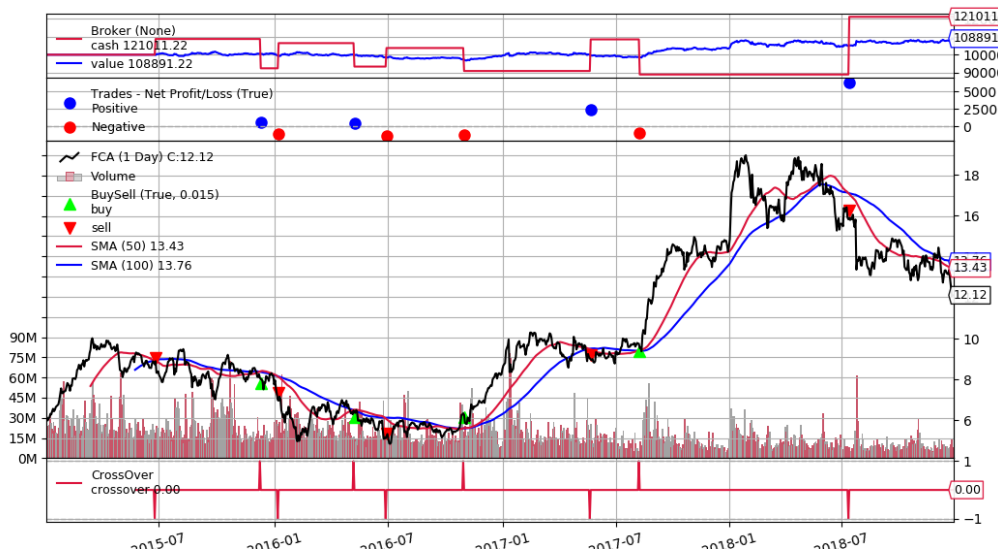
Graficare i dati e costruire indicatori



Graficare i dati e costruire indicatori

- Medie Mobili
- Medie Pesate
- Medie Esponenziali
- Bande di Bollinger
- RSI
- Tether Line
- Aroon Indicator
- Commodity Channel Index
- Hull Moving Average
- Volatilità storica di un titolo
- ...

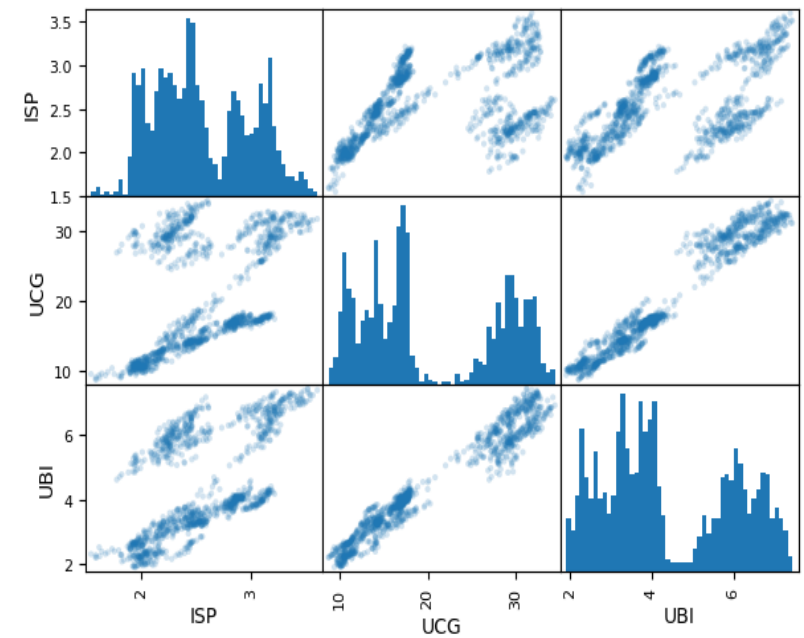
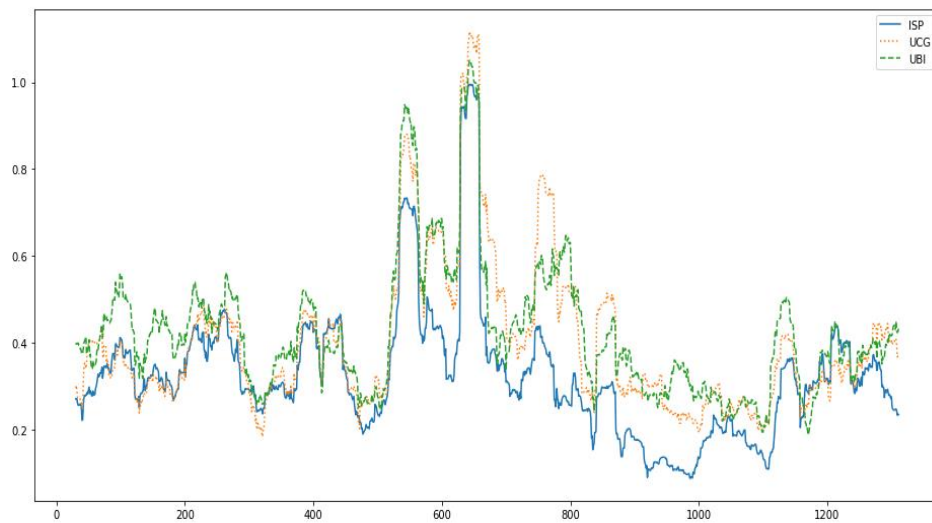
Trading System ed Ottimizzazione



3 # Analisi e Simulazione

Studio rischio e correlazione titoli

ALGOCAPITAL.IT



Analisi portafoglio e Simulazioni

