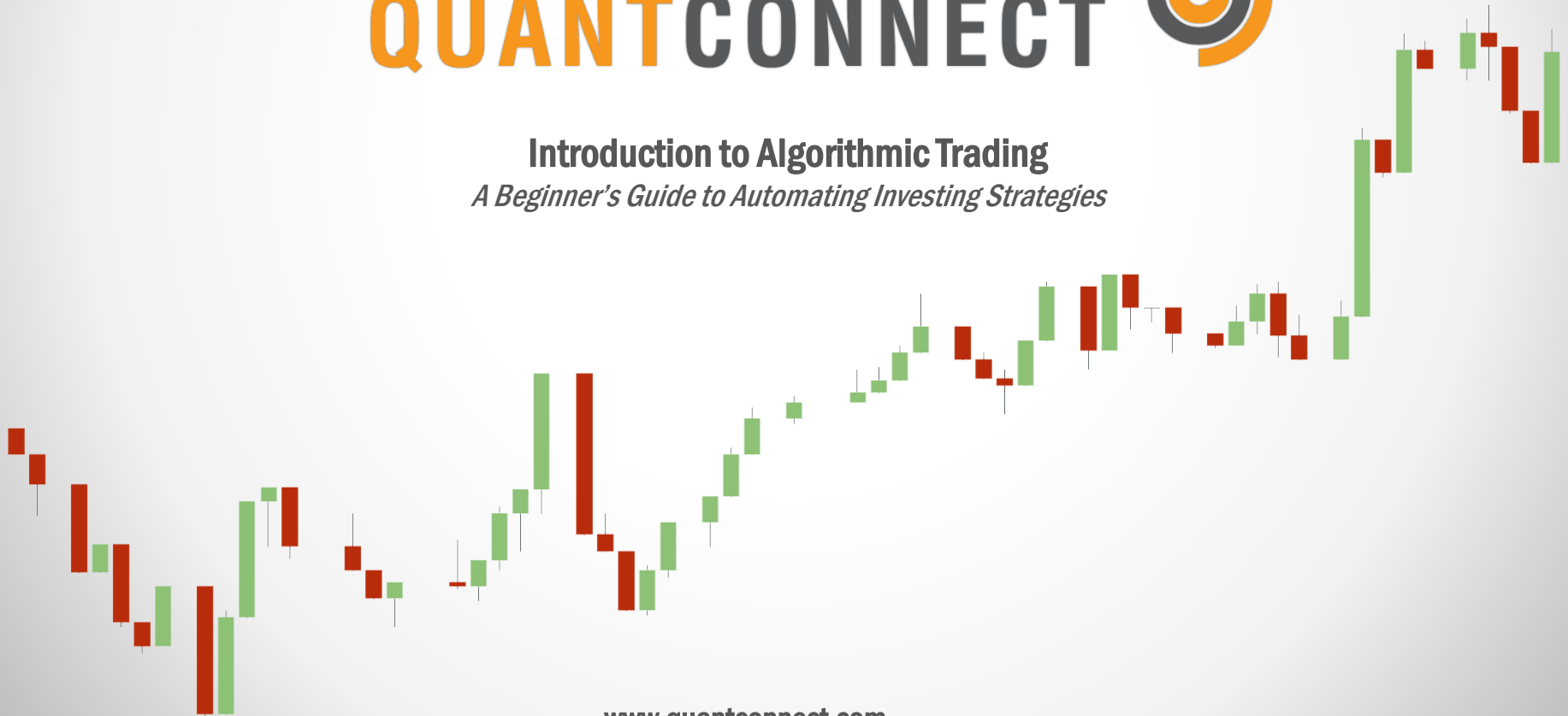


QUANTCONNECT

Introduction to Algorithmic Trading
A Beginner's Guide to Automating Investing Strategies

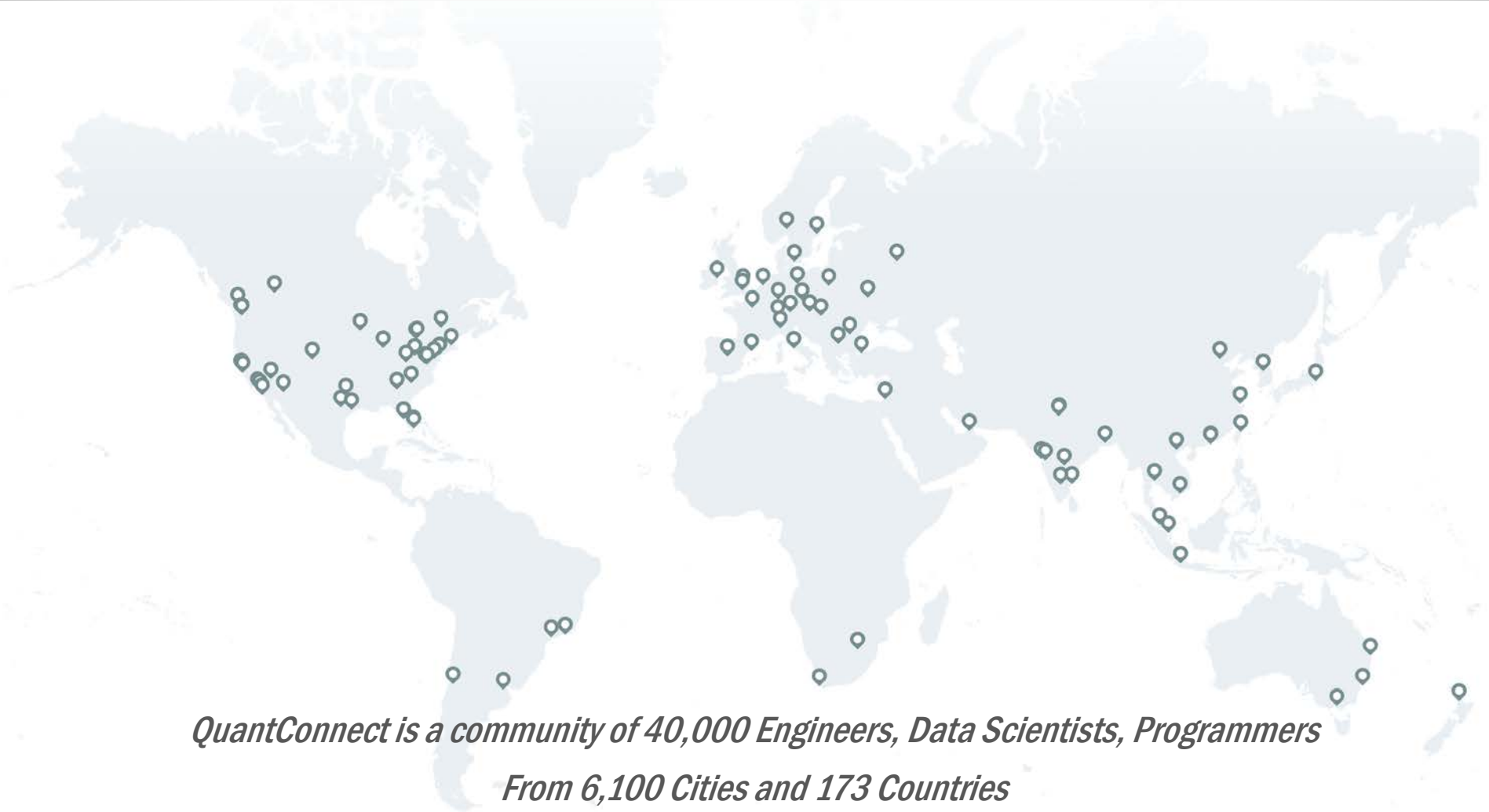


- What is QuantConnect?
- What is Algorithmic Trading? How can it help?
- Algorithm Development Process
- Creating An Investment Hypothesis
- Testing Our Hypothesis with QuantConnect Research
- Coding and Backtesting A Strategy
- Deploying Live

What is QuantConnect?



*We empower investors with powerful
investment tools and connect the brightest
minds from around the world with capital they need.*



*QuantConnect is a community of 40,000 Engineers, Data Scientists, Programmers
From 6,100 Cities and 173 Countries*

```

class ChewHistory(QCAlgorithm):
def Initialize(self):
'''Initialise the data and resolution required, as well as the cash and start-end dates
for your algorithm. All algorithms must initialized.'''
self.SetStartDate(2011, 01, 02) #Set Start Date
self.SetEndDate(2016, 12, 23) #Set End Date
self.SetCash(100000) #Set Strategy Cash
self.symbols = ["SPY"]
for s in self.symbols:
self.AddEquity(s,Resolution.Minute)
self.window=128
self.SetWarmup(self.window)
self.calc_len=self.window
for i in range(16):
self.Schedule.On(self.DateRules.EveryDay("SPY"), self.TimeRules.AfterMarketOpen("SPY",140+i*15), Acti
def runAndTrade(self):
# wait for warmup
if self.IsWarmingUp: return

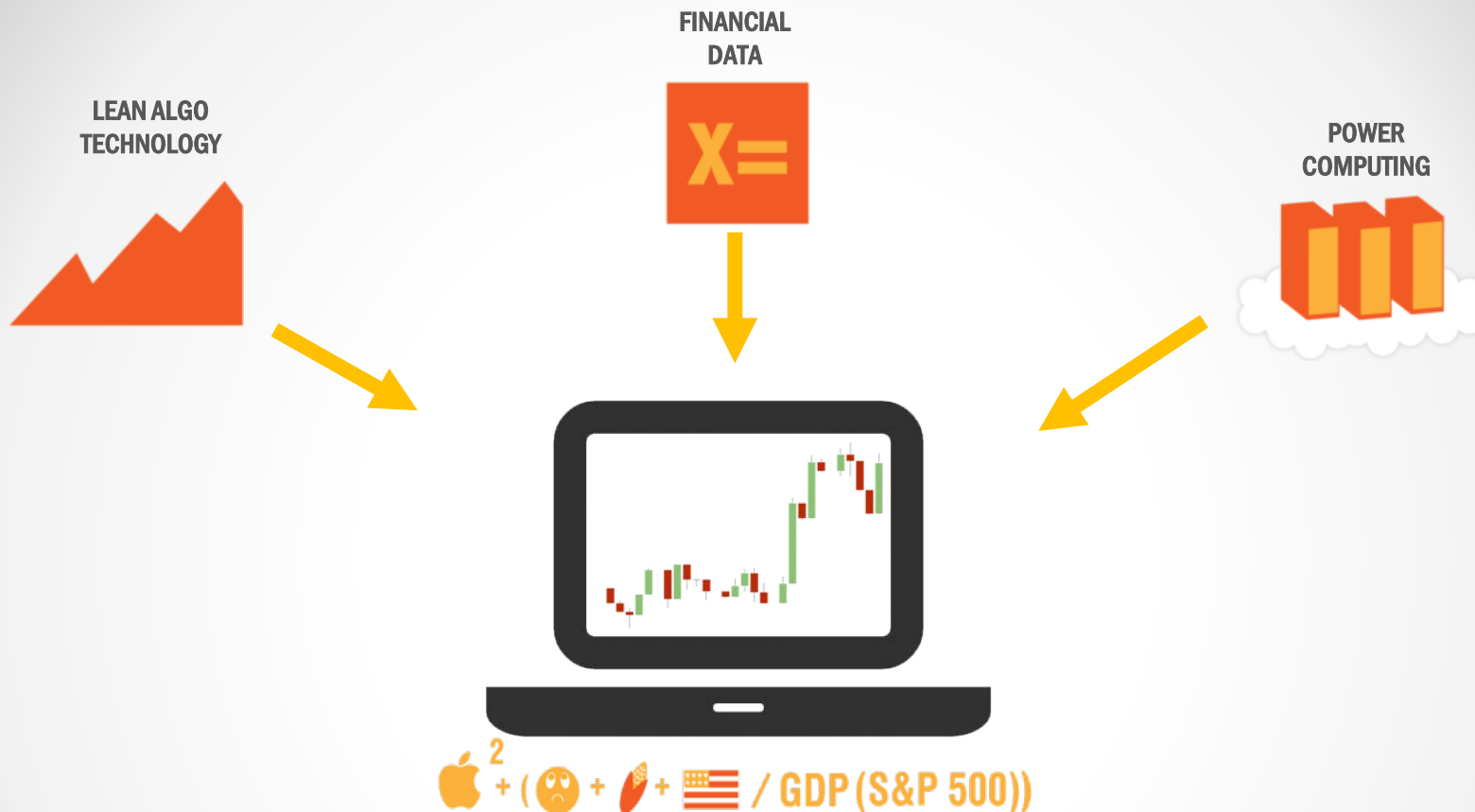
for s in self.symbols:
history = self.History(s, self.window, Resolution.Minute)
data=[]
index=[]
for slice in history:
data.append([np.float(slice.Open),np.float(slice.High),np.float(slice.Low),np.float(slice.C
index.append(slice.Time.date()) df=pd.DataFrame(data,columns=['o','h','l','p','v'],index=pd
if len(df)==self.calc_len:
self.Debug("Acquired "+str(len(df))+" rows of history")
r=df['p'].diff().fillna(0).values
a=np.diff(r)
m1=(r[-15:].mean()-r.mean())/r.std()
m2=(a[-15:].mean())/a.std()
#self.Debug("Finished calculations")
signal=np.sign(m1)+np.sign(m2)
self.SetHoldings(s,signal/self.Securities.Count)

```

Building Thousands of Algorithms Every Day



How do we do it?



We've built a web algorithm lab where thousands of people test their ideas on financial data we provide; for free.

$$\text{Apple}^2 + (\text{Sad Face} + \text{Corn} + \text{USA Flag} / \text{GDP (S\&P 500)})$$

- Using code for the implementation of your investment strategies.
- Wide variety of investment time horizons.
- Using math or scientific techniques in the investment process.

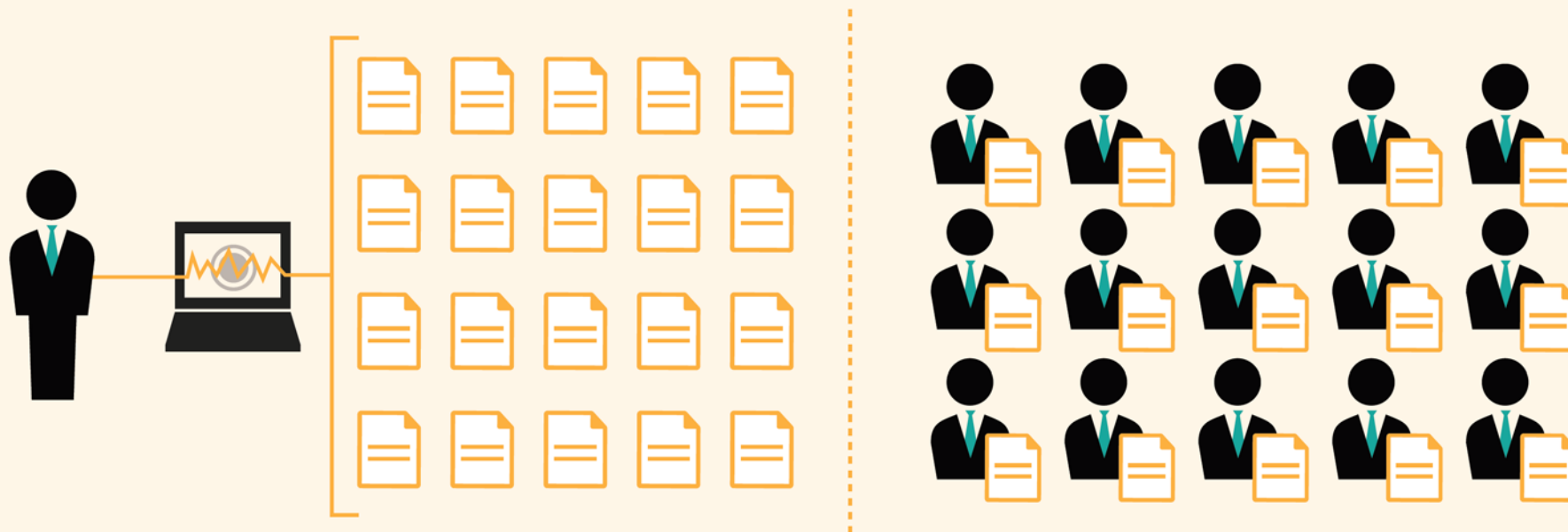
What Algorithmic Trading Is *Not*

- Get rich quick vehicle, or quick pathway to retirement.
- Not Solely HFT. *High Frequency Trading is one type of algorithmic trading.*



Process More, and Faster

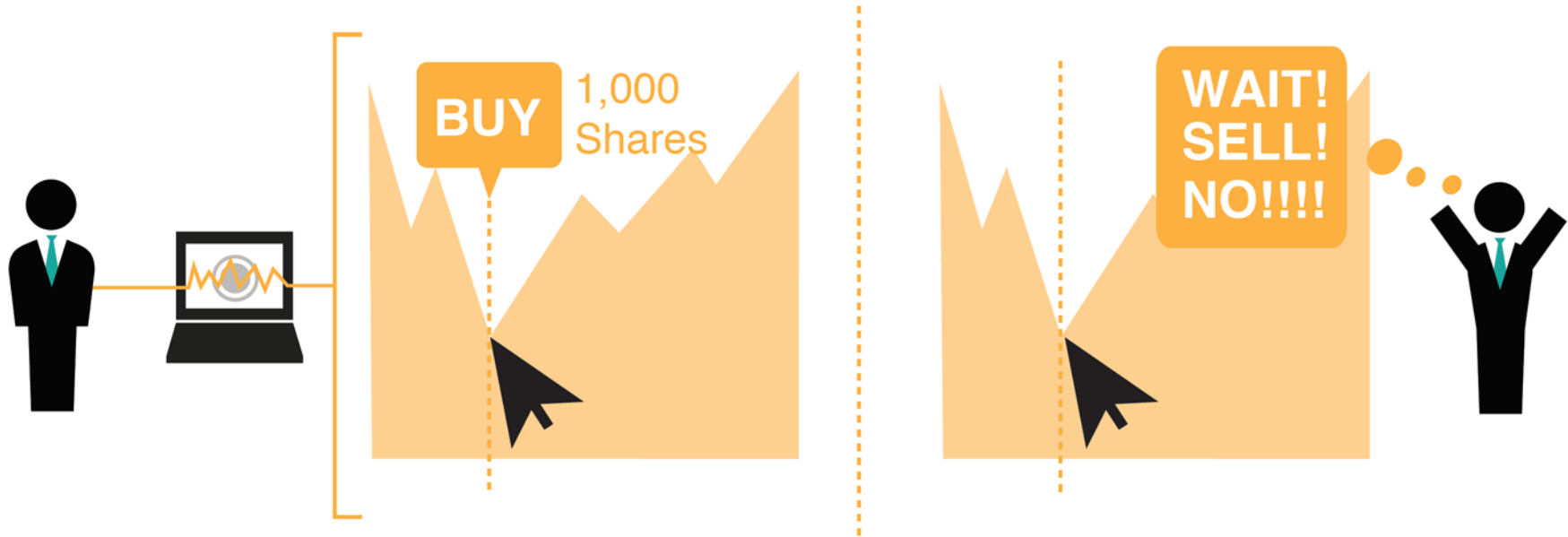
Algorithms take advantage of computers' ability to process data. They can compare price and metrics of millions of financial instruments faster than individuals.



#1. Using computers to do things people find boring and time consuming.

Emotionless

Emotions can get in the way of sound investing, causing many to buy when prices are going up and sell when they are going down.



#2. Enforce discipline in your investment strategy through automating execution.

Execute Faster

Algorithms are operated on computers, which react faster than humans to market changes.



There and back in **55.7 milliseconds**



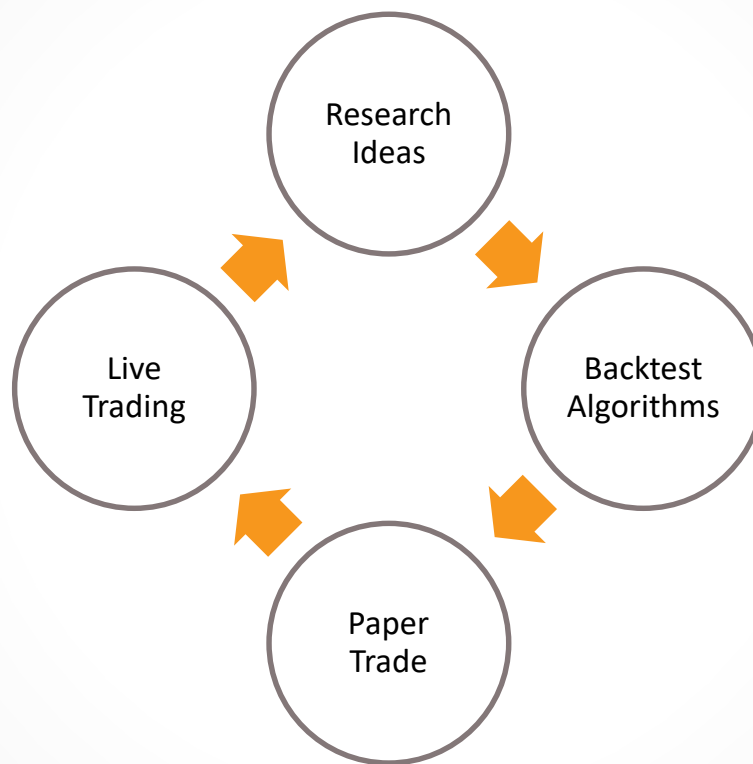
2x faster than the average blink



#3. Monitor portfolio 24/7 and control your risk.

Research

Quickly test ideas in a command line environment.



Live Trade

Deploy the strategy live in a real money account.

Backtest

Codify and run full simulation in the Algorithm Lab.

Paper Trade

Run idea on live market data.

Examples of an Investment Hypothesis

- Will rise of electric cars will make solely traditional manufacturers fall in value?
- Does the elected political party impact market stability, dynamics?
- Does consumer discretionary income and savings indicate better retail sales?

For Our First, Lets Keep It Simple

- The markets are smoothest when they're going up; and higher volatility is normally from panic. Should we wait on sidelines when there's panic?

Research Environment - x

QuantConnect Corporation [US] | <https://www.quantconnect.com/research/notebook/106/MyResearch.ipynb>

QUANTCONNECT Research BETA Pricing Data Community Algorithm Lab Research Documentation Jared Broad

File Edit View Insert Cell Kernel Help Not Trusted Python 2

```
In [14]: # For EURUSD
bbdf = qb.Indicator(bb, "EURUSD", 360, Resolution.Daily)
bbdf = bbdf.drop('standarddeviation', 1)
bbdf.plot()
```

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7efce87ea250>

In []: # Example with ADX, it is a bar indicator
adx = AverageDirectionalIndex("adx", 14)
adxdf = qb.Indicator(adx, "SPY", 360, Resolution.Daily)
adxdf.plot()

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7efce8439650>

Basic Template (C#) - Qo - x

QuantConnect Corporation [US] | <https://www.quantconnect.com/terminal/#open/675431>

QUANTCONNECT Terminal Pricing Backtest Data Community Algorithm Lab Research Documentation Jared Broad

Build Backtest Go Live Results

Basic Template (C#)

University

My Projects

Co-Pilot

Data Manager

Live Projects

Support

Files

Main.cs

+ Add New File

Libraries

+ Add New Library

General

Share

Clone this Project

Project Language: C#

Lean Engine: Master (5632df4b5)

Algorithm Parameters

+ Add New Parameter

Live Logs

Backtest Logs

Time	Log
2017-09-04 15:34:40	22274c8e08da3a8948718d09988061c6d
2017-09-01 21:36:41	'6153598a08d5629363e447c7d541ac1
2017-09-01 21:35:47	9a0b6018c2c575f06fb9e8330665ae5a
2017-09-01 21:35:37	1614aa7a5b09032cf2e2964fa1c2888
2017-09-01 21:25:38	80dae2fa881f3a8832c8b44b62d5e283
2017-09-01 21:25:03	90f87282d971337c2959ccc0ae034464
2017-09-01 21:20:02	598b97b7770141340eb0ab72e6c8965
2017-09-01	751869f1d2057e7d6dda8562b0c53008

```

33 private List<String> _universeSymbols = new List<String>();
34
35 public override void Initialize() {
36     // this sets the resolution for data subscriptions added by our universe
37     UniverseSettings.Resolution = Resolution.Second;
38     UniverseSettings.FillForward = true;
39
40     // set our start and end for backtest mode
41     SetBrokerageModel(BrokerageName.InteractiveBrokersBrokerage);
42     SetStartDate(2017, 08, 18);
43     SetEndDate(2017, 08, 24);
44
45     var spyEquity = AddEquity("SPY", Resolution.Second, Market.USA, true, 0, true);
46     // Schedule events
47     Schedule.On(DateRules.EveryDay("SPY"), TimeRules.AfterMarketOpen("SPY", 5), () => {
48         OpenPositions();
49     });
50     Schedule.On(DateRules.EveryDay("SPY"), TimeRules.AfterMarketOpen("SPY", 15), () => {
51         ClosePositions();
52     });
53     Schedule.On(DateRules.EveryDay("SPY"), TimeRules.BeforeMarketClose("SPY", 1), () => {
54         canPoll = false;
55         List<OrderTicket> stopTickets = Transactions.GetOrderTickets(x => x.OrderType == OrderType.StopMarket).ToList();
56         foreach (OrderTicket stopTicket in stopTickets ?? Enumerable.Empty<OrderTicket>())
57         {
58             Debug(String.Format("Cancelling stop order for {0}", stopTicket.Symbol));
59             stopTicket.Cancel();
60         }
61         Debug("Finished Cancelling stop orders for the day");
62
63     De
64     }); Debug
65     // DebugMessages
66     Add DEMA
67     {
68         define local
69         Debug local
70         decimal local
71         Decimal local
72         default keyword
73         using (var client = new WebClient())
74         {
75             // handle live mode file format
76             if (LiveMode)
77             {
78                 // fetch the file from dropbox
79                 String file = client.DownloadString(url);
80                 // if we have a file for today, break apart by commas and return symbols
81                 if (file.Length > 0)
82                 {
83                     file = file.Replace("\r\n", "");
84                     _universeSymbols = file.ToCsv();
85                     _universeSymbols.ForEach(Debug);
86                     return _universeSymbols;
87                 }
88             }
89             // no symbol today, leave universe unchanged
90         }
91     }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

Unsaved. C#

Going Live, Deploying to Live Trading

Live Clone of FX Comb... X

QuantConnect Corporation [US] | https://www.quantconnect.com/terminal/#openLive/615470

Terminal Upgrade Mode **Live** Data Community Algorithm Lab Research Documentation Bernard Blazquez

Dashboard \$2.21 Unrealized -\$0.00 Fees \$0.00 Net Profit 2.81 % Return \$132.08 Equity \$132.10 Holdings \$259.79 Volume

Zoom 10m 1h 1d 1w All

Jul 26, 2017
Equity: \$131,444.5

Market Status Time
Equity Open 18:24:02 EDT
Forex Open 18:24:02 EDT

Select Chart
Strategy Equity
Benchmark
Select Stockplot

Status
Running

Server Statistics
CPU: 79.1%
RAM: 320 MB / 512 MB
Host: QC-COLO-1-9f0ca110
Up Time: 57d 01:35:09

Clone of FX Combined Momentum and Correlation
Algorithm: 11711230011773

Symbol	Average Price	Quantity	Market Price	Market Value	Unrealized
GBPUSD	\$1.29897	100	\$1.32102	\$132.102	\$2.20

Orders										
Open Orders Only										
Id	DateTime	Symbol	Fill Price	Quantity	Type	Operation	Status			
4	2017-09-01 09:31:05	USDCAD	\$CAD	-136	Market	Short	Invalid			
3	2017-09-01 09:31:05	GBPUSD	\$1.29897USD	100	Market	Long	Submitted			
2	2017-08-01 09:31:05	USDCAD	\$CAD	-132	Market	Short	Invalid			

Console