Why Your Algorithm Will Fail

Reason 3: The Optimization Process

> ***"I fear not the man who has practiced 10,000 kicks once, but I fear the man who has practiced one kick 10,000 times."***
>
> **- Bruce Lee**

Before you continue -

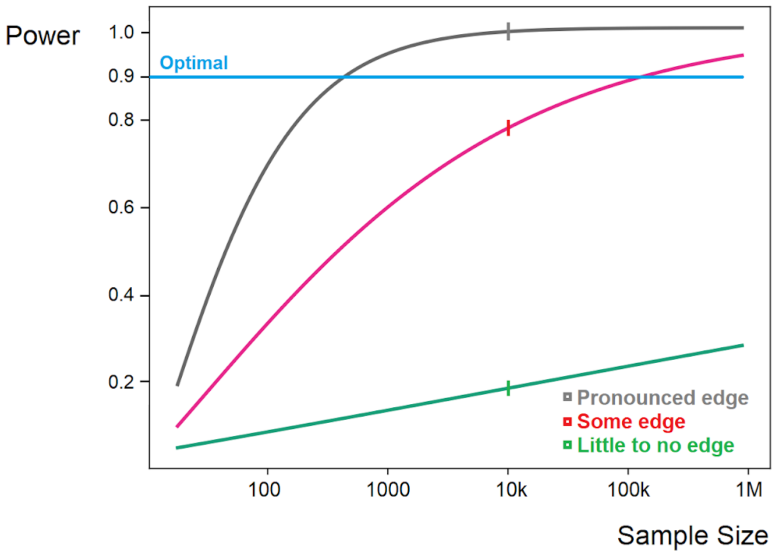
If you want to go any further as an algorithmic trader, then it's time to get serious. **Stop treating algorithmic development like a hobby and start treating it like a profession.** To be an alpha trader at the top of your game, you need to behave like a ***statistician*** in order to understand how to plan your studies and optimizations. You need to be a ***data scientist.*** What you do is a ***science.*** And as we all know, **science is based on experiment, not dogma.**

Carry on.

As it turns out, Bruce Lee was onto something. There's a concept in statistics known as **statistical power analysis.** For **quantitative traders,** such as ourselves, it is a tool we use to answer one very simple question: Does a system have an observed **edge?**

New term alert: What is an **edge**?

An edge is simply a system's capacity to gain more than it loses *over a statistically significant sample size*.



Statistical Power Analysis

The level of power most statistical studies aim to achieve is around 0.9.

By the time you have 10,000 samples, you should be able to tell if a system has a pronounced edge.

Statistical power itself is the ***probability*** of detecting an edge if an edge is present. If a system demonstrates an edge, we can be reasonably certain (>90%) that edge is true after **no less** than 5,000-10,000 samples. We have already learned that an **overfitted** system ***looks*** like it has an **edge,** but in reality is just a **random, specific,** system that is **unable to generalize**. But if a system looks great in backtesting, how can we tell if it's a piece of crap or not? Well, we need to look under the hood. How was the system built, and **most importantly, what was the *sample size* used?**

As we'll find out, this is actually a double-ended question. **What we *really* need to know is: how many in-sample and out-of-sample trades does this system have?** If I were a betting man, and I am, I would bet that you have probably never used **out-of-sample** data in your backtesting - only in your **forward testing** (if that). Some YouTube stars make no mention of sampling, and some simply suggest trading an algorithm for something like 6-12 months to see if it works. Great idea, right?

Not quite. Not only is this time period generally inadequate for establishing reliable statistics for your system, it's also a horribly inefficient use of time. What happens when your **overfit** system fails, like it almost certainly will? What if it takes 6 months, a year, or even longer? Do you go back to the drawing board and watch your *next* **overfit** system shit the bed for 6 months or more?

Absolutely not. This is a colossal waste of time, and can actually be accomplished during backtesting. I'll explain how this can be done later in the article, but first, let's get back to the **sample types.** It's important to understand the definition and purpose of each sample type, because you'll be employing both of them like a **statistician.**

We'll start with the definitions:

In-sample (IS) data: A sample that occurs **during** your parameter optimization.

Out-of-sample (OOS) data: A sample that occurs **outside** of your parameter optimization.

In simple terms, IS data is for *extracting* parameters, and OOS data is for *testing* those parameters on data they haven't seen before.

Let's re-introduce our plumber. In light of his recent unemployment and divorce, he has put all his effort into acquiring a time machine - and succeeded. Now, we travel back in time to find out what went wrong with his wrench selection.

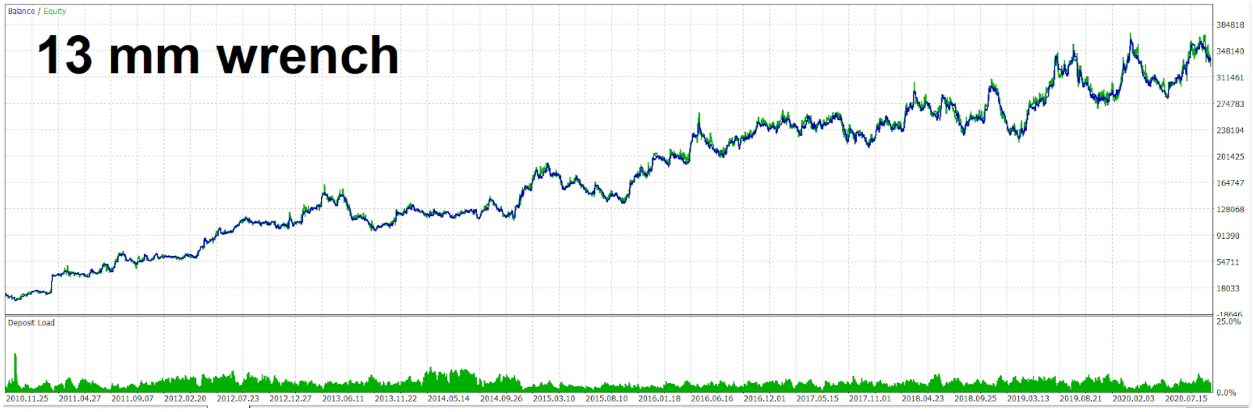
3…

2…

1…

!*!*!*TIME MACHINE NOISES*!*!*!

You (the plumber) have arrived exactly 10 years ago, back to the first day you started working in the industry. You initiate the fast forward function on your time machine while noting the wrench you used for each job, and arrive back in present time. Over 10 years and 10,000 jobs, you found that you actually used the **13 mm wrench** the most - not the 10 mm. Maybe this was where you went wrong - the 13 mm was the best choice all along! Right? It certainly has a good-looking **equity curve.** It's showing over **3000% returns!**



Well, given your newfound understanding of **overfitting,** you know better than to fall for this one again. You *know* that just because that **specific** 13 mm wrench worked the *most* for whatever reason, doesn't mean it will be the *best* choice for your next job, or any job thereafter - especially if you're working in uncharted territory. **But *why* does that equity curve look so damned good?**

You need to think outside the box to answer this one. You step in the time machine once again and go back 10 years. This time, while traveling through the wormhole, you are approached by an interdimensional being named **Goku:**

“Kaaaaaaaameeeeeeee - haaaaaaameeeeeeeee - haaaaaaaaaaaaaaa!!!!!”

Interestingly enough, you understand exactly what he said. You give the Saiyan a nod and thank him for answering your question, as he returns to kick ass in the Tournament of Power. You arrive back in time again and, armed with new divine knowledge, you are ready to approach this optimization like you never have before -

Flashback: let’s review how you currently build systems. You probably optimize over a 2-5 year period, take the “best” performing parameters and trade them, right? But by now, you know that’s a mistake. The beautiful equity curve you just saw is an **overfit** lie. Currently, the only way you know how to find out if that wrench is **robust** or **random** is by forward testing.

We don’t have time for that anymore. Now it’s time for you to meet the new way of testing. Welcome to the 21st century, caveman. Let’s resume your new test -

You input the **same test parameters that gave you the 13 mm wrench.** 3 years pass, and - **PAUSE!** You stop the flow of time -

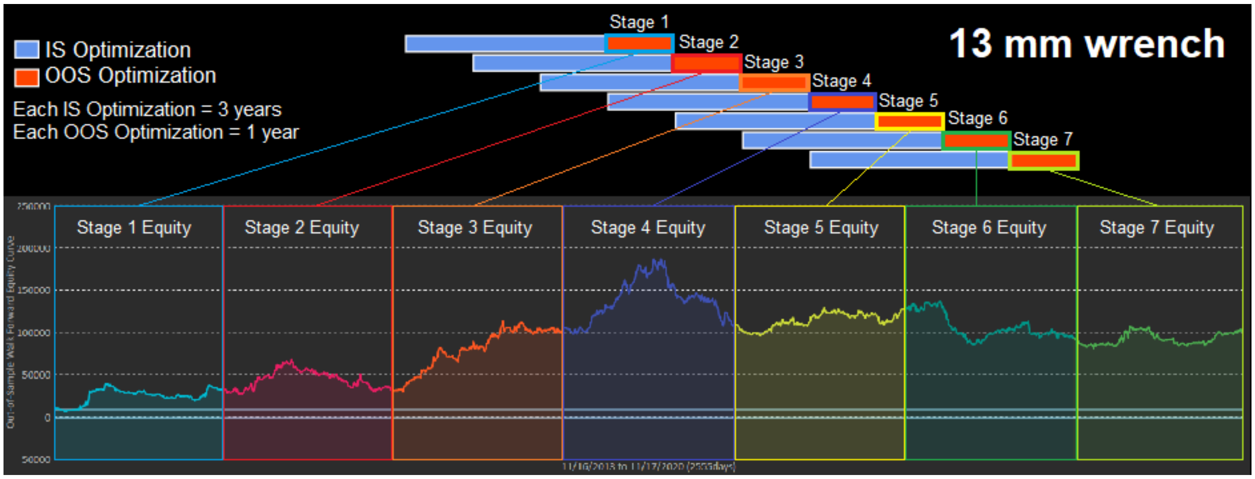
Now, instead of continuing the backtest, you *forward test* the parameters you extracted from the IS data over the next year of history - the OOS data.

You <u>disregard</u> the equity curve from the IS optimization.

You <u>record</u> the equity curve from the OOS optimization.

Read that again until you understand it.This process is called **Walk Forward Analysis (WFA).**

You then rewind and repeat, as you construct your fancy new **OOS equity curve:**



What a difference! That can't be the same system we saw before, **can it?** Well, it is. So now we know the **IS equity curve** and **OOS equity curve** are from the **same system** (Yes, they **REALLY** are. It just goes to show how **INSANE** overfitting can look). But why do they look so different?

Well, one of the ways **IS optimizations distort results** is by rewarding & compounding lucky early performances. As you can see in the **OOS equity curve,** the 13 mm wrench performed quite well for a few years, then stopped demonstrating an **edge.** By the time the strategy failed, the system already compounded a lot of profit. An **IS optimization** would have led you to believe it was a great system, when it was actually inconsistent.

With this in mind, we have learned that **WFA** decides if parameters extracted from **IS data** would **actually perform on OOS data.** It is a **true representation** of a system's **robustness** or lack thereof.

In a nutshell: WFA simulates live trading (if done properly). Now, by the end of the 10 years, you now have about 7,000 **OOS data** samples to look back on. Contrary to the **IS equity curve,** you find that **none** of the wrenches performed very well. They were not **robust** choices. If you chose to trade with the system represented by the equity curves, **you would have entered the market with a strategy that hasn't performed in years.**

Now, the nature of the problem has changed considerably: instead of trying to build great **IS equity curves,** your focus shifts to building **robust OOS equity curves.** Easy enough, but what do we mean by **robust?**

Robustness is simply a system's capacity to adapt to changes in live market conditions *over a statistically significant sample size*.

We want to see consistent, upward slopes on OOS data. This indicates **robustness.** It is impossible to determine **robustness** without **WFA**, unless you're willing to wait a LONG time. In fact, **WFA** is superior to traditional backtesting and forward testing for 3 main reasons:

1. You're **much** less likely to **overfit** results when you use **OOS data** to build equity curves.

2. You can **quickly** decide if a strategy is **robust** instead of waiting months or years.

3. You will **never** go back to trading with strategies that lack a real **edge.**

I know this is probably a lot to take in, so feel free to re-read the previous sections. This is the most technical article in this series, so once you understand these concepts, it's smooth sailing from here on out.

This should qualify as one of the most eye-opening but mild **Dunning-Kruger** episodes you've ever had. I'm sure of it.

So now we know about **overfitting,** and we know how to prevent it through **WFA.** We have learned the difference between **in-sample** and **out-of-sample** data, and how to use them to **structure an optimization process.** Perhaps we can save your future plumbing career and marriage after all! -

Right?

Not yet. First, we have to address another component of optimization. It's something that is widely misunderstood, but is arguably the most critical component of efficient algorithm development -

The optimization approach (you will love this one).