Kevin Stone

1 Follower

The 5 Most Profitable Rules in Blackjack

Kevin Stone 1 day ago · 15 min read

A comprehensive, Python-powered simulation study into the practical probability of optimal Blackjack strategies

Photo by Michał Parzuchowski on Unsplash

Blackjack. A staple of 52-card games, and a flagship table destination for casino gamblers. Start with 2 cards and beat the dealer to 21 — it sounds so simple, doesn't it?

Not only is it one of the most approachable games in the casino, but it famously boasts one of the highest returns to the player. Understandably, Blackjack finds its home as one of the most popular games-of-chance, with that not-yet-insubstantial shimmer of hope for padding your wallet.

And yet, the house always wins.

> *Because the House always wins. Play long enough, you never change the stakes, the House takes you. — Danny Ocean, Ocean's Eleven*

That return to the player, as high as 99.8% in some variations of Blackjack, is still less than 100%. Any given hand, you are *expected* to lose money. Of course, that's the long-run expectation — and who has time to play an infinite number of hands? When you step up to the table, you're going to be in that 20-something% population that walks away with a few extra bills, aren't you?

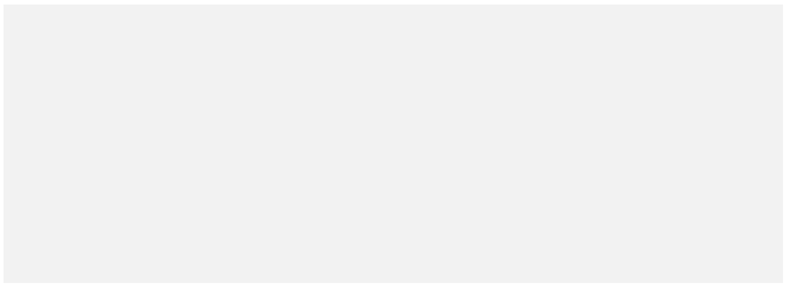
After all, sometimes you can even buy an "Optimal Strategy" cheat-sheet in the casino gift shop. Follow the guidance on the flashcard, and you're guaranteed the best possible return — Bob's your uncle!

But where does that optimal strategy come from, and how does it work? What if you can't bring the card to the table, Google it while playing, or keep a few dozen rules casually on the front of your brain after a few libations?

That's the rub — and we'll explore it here using a simulation-based approach in Python, based on a Master's program project of mine.

Let me show you how and why you can capture more than **90% of the value of full optimal strategy** at a mere fraction of the mental cost, with only **5 easily-memorized rules**.

Further, the impact of my simplification to that short-term edge we are chasing is practically *indistinguishable* by many measures. Take that to the bank!



Preview of the 5-rule Minimalist Optimization — read more below!

· · ·

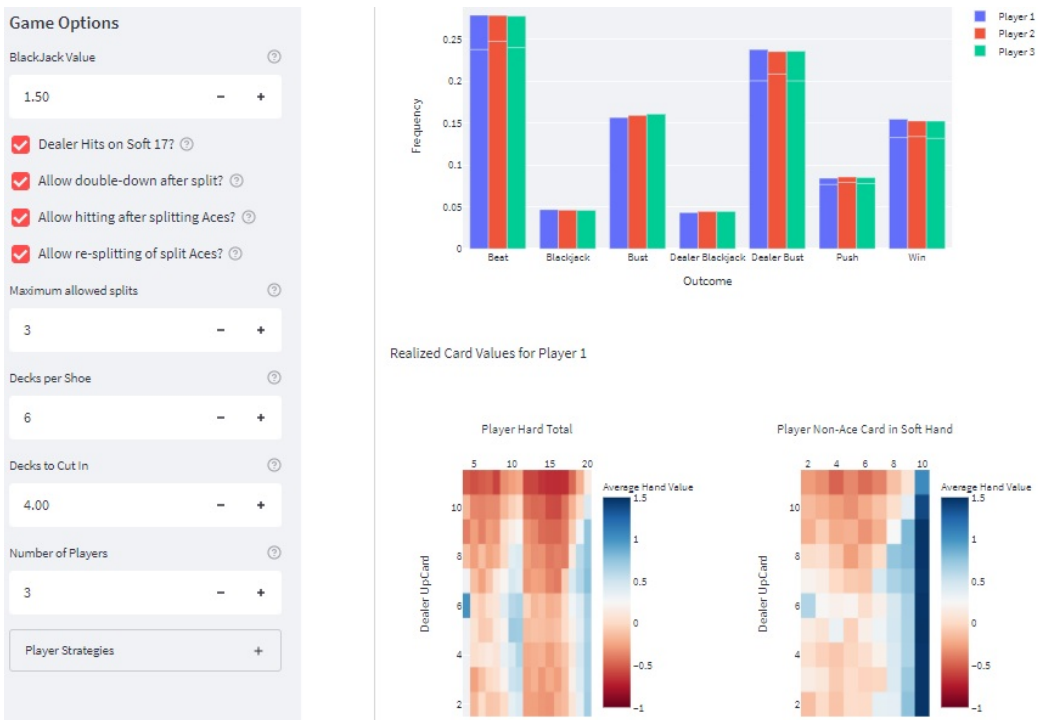
Suit Yourself

Before I describe what I've learned along the way, I would encourage you to visit the web application I've developed using Streamlit.

```
import streamlit as st
```

Here, you'll find all of the knobs and levers you need to explore the game on your own — rediscover my insights and push beyond them.

Interactive Blackjack Strategy Simulator (via Streamlit & Heroku)


Game Options

BlackJack Value: 1.50

- [✓] Dealer Hits on Soft 17?
- [✓] Allow double-down after split?
- [✓] Allow hitting after splitting Aces?
- [✓] Allow re-splitting of split Aces?

Maximum allowed splits: 3

Decks per Shoe: 6

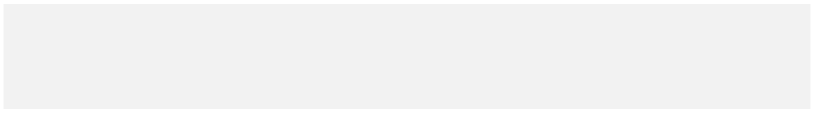
Decks to Cut In: 4.00

Number of Players: 3

Player Strategies

Sample results from the interactive strategy simulator, and a preview of the game-options sidebar

Hopefully, you’ll find the interface easy to use and the figures insightful. Share it with your friends, or pay a visit before your next card game. Play with any combination of common rule-changes or variations of optimal strategy. Take it a step further, and test drive your own purely custom strategy!


Favorite feature: The action analyzer — not only what is the optimal action, but by how much, and why?

Take your analytics to the next level with card-value heatmaps, hand-outcome breakdowns, and distribution analysis.

Perhaps above all, a valuable feature is the Player Action Analyzer. On no short of hundreds of websites, you can find basic and optimal strategy — but no where else can you optimize it yourself *LIVE* according to so many rule combinations! What's more, don't just stop at the "best" action, but evaluate how much better it is than the alternatives — and understand why by comparing the distribution of outcomes.

When you're satisfied with your own tinkering, take a trip back here and let's compare notes.

. . .

The Game in Code

Blackjack is played in essentially 5 repeating steps including (1) shuffling the deck, (2) dealing the cards, (3) declaring the dealer "upcard" and checking for dealer Blackjack when necessary, (4) allowing each player to act on their cards, and (5) resolving each hand and bet.

I chose to automate this logic using three classes: `Game(options)`, `Player(role, strategy, rules)`, and `Deck(decks_per_shoe, cut_in)`. The core 5 steps above run in a method `Game.play()` which interacts with such sensible class methods including `Deck.deal()` , `Player.decide()`, `Game.act()`, and `Game.resolve()` as highlighted in the code snippet below.

The currencies between these key methods are the `player_sum` of Player's card values and the Dealer's showing `upcard` from which the decisions, actions, and resolutions are derived. The function below depicts how such a sum is calculated, where the presence of multi-value Ace demands a condition for altering sums that would otherwise exceed 21 and lead to a loss. This condition is not as *decision-based* as some descriptions imply; but rather, algorithmic in a way that provides a correct sum or a failing sum and applies equally for both player and dealer.

With a player sum and dealer card known, each player is faced with a decision — to **Hit** (draw 1 card with the option for further draws), **Stand** (do nothing) , or **Double Down** (double the bet and draw 1 card without the option for further actions)— with the ultimate objective of beating the dealer (sum closer to 21 without exceeding). Certain circumstances and rules allow for alternative actions **Split** (separate 2 qualifying equal-value cards into 2 independent hands that will be hit once each before continuing play separately) and **Surrender** (concede half of the initial bet to mitigate a likely loss of full value).

Optimal (a.k.a. *basic*) strategy is a universally-accepted set of actions that inform the statistically superior response to each decision in Blackjack assuming the distribution of cards in a full, shuffled deck. Two major variations exist for different popular versions of the game and many report conditional actions that handle minor variations in rules, or other dynamic elements of gameplay.

Although the dealer *must* always stand on totals of 17 or higher (hitting on less), the two major variations of the game demand the dealer hits (H17) on a *soft* 17 (Ace + other sum of 6) or stands (S17) on the soft 17. As a point of example, the optimal strategy for the more common H17 game with 4–8 decks is shown in the figure below.

Optimal Strategy for Blackjack (H17)

This strategy, and its analogues, can be broken down into three major parts with two variations: **Hard Total** strategy (any cards not including 11-value Aces), **Soft Total** strategy (any cards including exactly one 11-value Ace), and **Split** strategy with or without Doubling-Down and Surrender. We will explore how each of these discrete elements combines to provide the value of optimal strategy to the player.

The many variants of these different optimal strategies are loaded into `Player` class attributes and accessed by the `Player.decide()` method as detailed below. Hierarchically, we must first determine which actions are available based on the rules, then seek Split, Soft Total, and Hard Total strategies as allowable and in that order. If our strategy tables return a conditional action, we must also process these based on the rules or the dynamic conditions of the game (i.e. if double-down after split [DDAS] is disallowed and a decision must be made for a split hand).

As is shown in the `Game.play()` method, the `Player.decide()` and `Game.act()` methods exchange values recursively until the final set of decisions (for all hands in the player’s control) include only *Stands* or *Surrenders*. The `act()` method fulfills the requests of `decide()` and enforces a number of rules that are impractical within the `Player()` class.

Once every hand reaches a point that no further actions are decided and requested, the dealer reveals their hole card and acts according to the more rigid strategy described above. Fortunately, our `player.decide()` method handles this the same as any other player! Last, then, the `Game.resolve()` method handles the scoring of each hand. A delicate hierarchy must be maintained to properly simulate the true game, with 6 levels as expressed in the code snippet below. As one example, the condition of a player "busting" (exceeding 21) supersedes any other condition of the dealer's cards (including a bust) resulting in the loss of the player's bet.

It is also within the `Game.resolve()` method that I have chosen to store the records of each hand. From this attribute `Game.record_keeper()` we can use our typical data analytics toolbox to excavate the insights from within the depths of Blackjack rules and optimal strategies.

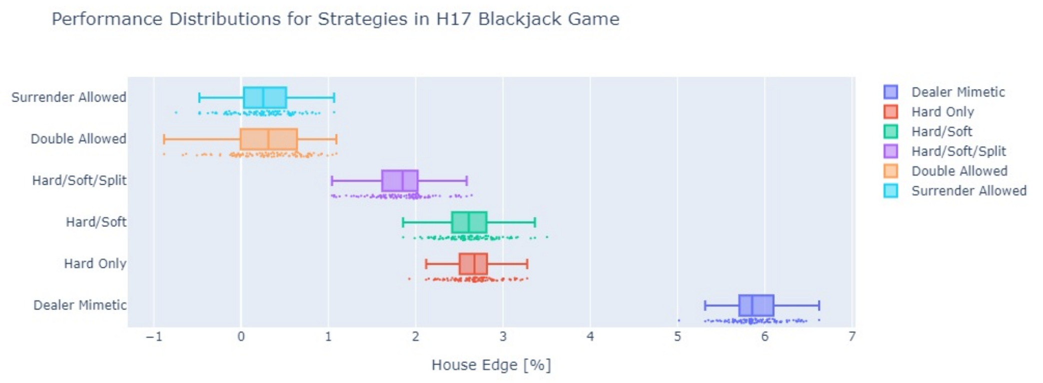
. . .

The Game in Numbers

In order to set up a statistically robust exploration, a fixed number of hands played is replicated a very large number of times (apropos *Monte-Carlo* simulation). If we want to understand the impact of strategy on average returns, as will often be my target, we will need to simulate both a large number of hands and a large number of replications so that the variance of the "batch means" does not cause overlap in the distributions of our sample sets. Where this has been the objective, I have in general simulated 100 replications of 100K hands for a total of 10MM individual hands resolved. In total, approximately one half of a billion hands were simulated to produce these results!

With this approach, let's begin by looking into the implications of strategic elements for optimal H17 and S17 gameplay as promised.

Shown below is a box plot for the H17 strategy broken down into pieces, with incrementally simpler strategies moving down the vertical axis. With surrender and double-down enabled alongside the other 3-piece decision tables for full optimal H17 strategy, we hone the house edge down to a mere 0.24%. That is, a player following this strategy is expected to keep (on average) 99.76% of their bet each time a hand is played. On the other hand, simply mimicking the dealer's actions yields a horrid 5.89% house edge explained in large part by the uneven disadvantage of the player to lose, even if both player and dealer bust.



Data generated from 10MM hands of H17 game with batches of 100K (DDAS/rehit-AA/resplit-AA allowed, max. 3 splits, 6 decks with 4 cut-in, 3/2 Blackjack)

As is evident in the figure and corresponding table below, most of the edge is reduced with the inclusion of Hard Total strategy and the enabling of Double-Down. Because of the relative rarity of drawing an Ace or double card, the Hard Total strategy informs the majority of player decisions, thereby yielding the most impact through sheer frequency. Doubling down imposes its benefit by increasing the yield of a bet in situations that are expected to be beneficial, stretching the value distribution forward to raise its mean.

Almost negligibly important (in the absence of doubling down) is the Soft Total strategy. As it turns out, this strategy appears less frequently and is relatively unimpactful when compared with the default dealer decisions. However, this is **NOT** the case for the S17 game, in which S17 optimal strategy for soft sums derives its value by emulating the H17 default strategy (hitting on soft 17s).

Data generated from 10MM hands of S17 game with batches of 100K (DDAS/rehit-AA/resplit-AA allowed, max. 3 splits, 6 decks with 4 cut-in, 3/2 Blackjack)

In either case, the inclusion of **Split** strategy can be considered the 3rd most important piece of optimality within the 5 divisions I have somewhat subjectively determined. Interestingly, the *vast* majority of optimal split plays have negative expected values. As such, the target has less to do with maximizing a positive outcome but more with minimizing a negative one — even on otherwise advantageous low dealer upcards (6 or less). This is the case with most split 2s, 3s, 6s, 7s, and especially 8s. Only double Aces and *some* 9s yield positive expectations.

If you’ve been following optimal strategy in the past only to be disappointed by consistently negative outcomes from the “perfect” recommended splits, fret not! Improving your edge has as much to do with diminishing the dealer’s advantage as it does improving your own.

As previously mentioned, there are many other minor variations to the rules of Blackjack beyond S17/H17. However, in large part these variations have very little effect on the house edge as summarized in the following table.

In general, these variations have little more than an absolute 0.1% increase or decrease to the base case we have been using. Of note, however, is the dramatic importance of the natural Blackjack bonus value. Games which offer less than a 3/2 payout on Blackjack are absolutely best avoided. Apart from this, players should also watch for games that disable the re-hitting of split aces and doubling-down after splits, both of which also noticeably damage the player’s edge.

. . .

What’s In It For You

Depending on how you count, getting from dealer strategy to optimal strategy involves 24 rules. How *I’m* counting is based on contiguous positions on the strategy table (combinations of player cards and dealer cards) which share the same optimal actions. These lead to 24 clusters of rules that can be written using *AND* or *BETWEEN* logic but without *OR* statements).

That is a perhaps unreasonably large number of rules to memorize for a casual night at the casino!

Forget all about that!

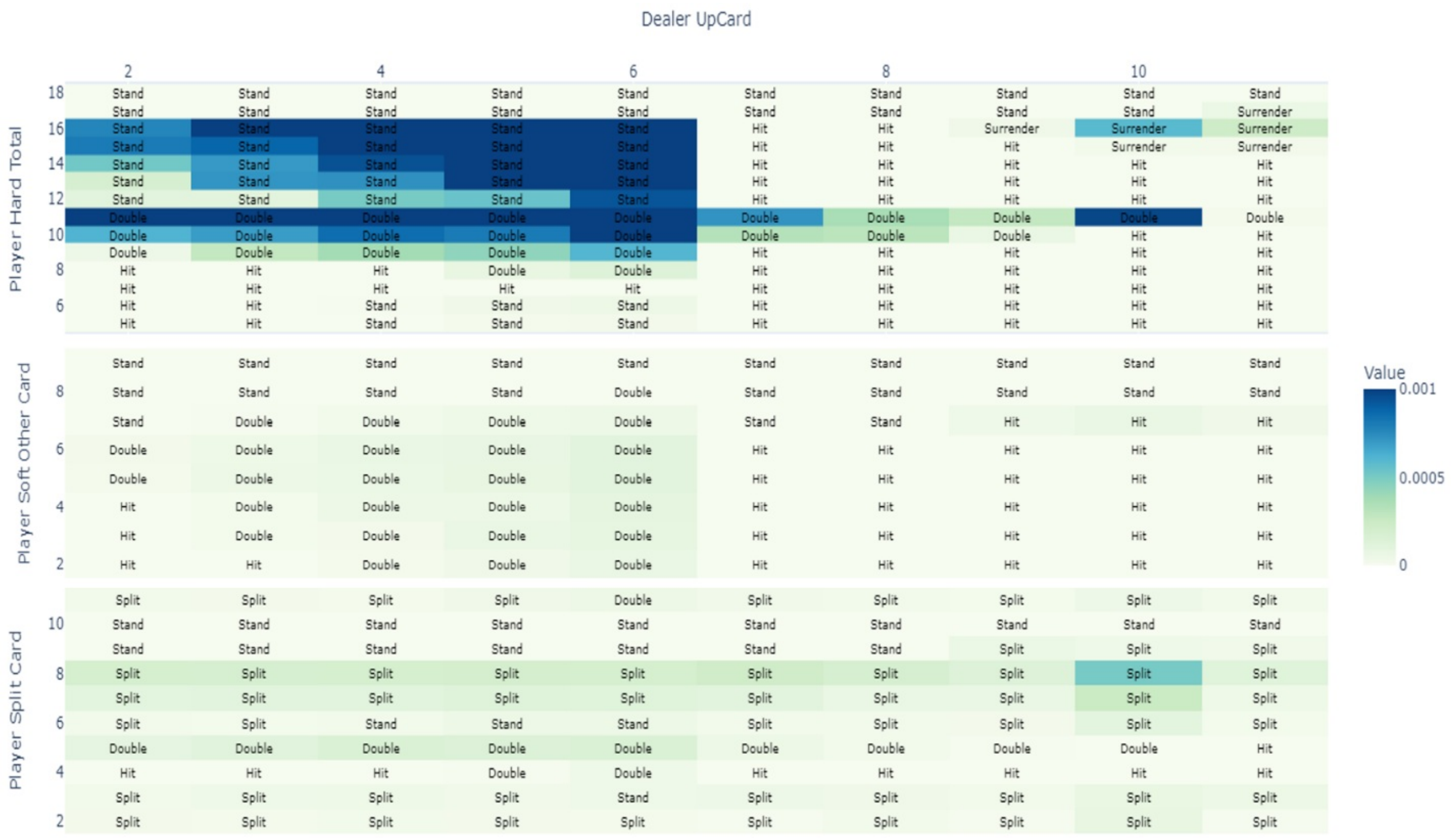
Using the techniques described above, albeit applied in a *slightly* different manner, we can answer the important question: **which are the most important rules to memorize?**

Let’s simulate every possible combination of player cards and dealer upcard, then randomly sample dealer hole cards a large number of times. For each iteration, we’ll force the player to take a particular *first* action and then use the default dealer strategy to inform any future actions if necessary (e.g. for splits or re-hits). We can then examine the average expected value for the full set of potential first actions at each opportunity in the strategy tables.

The difference between the best action and the otherwise default dealer’s action (a trivially memorized strategy of 1 rule) is therefore the **differential value** of the optimal action. That’s a great start, but not all opportunities on the table are created equal. As hinted before, the probability of needing a hard-total strategy is much higher than that of needing the soft-total or split strategy. For example, there are *far more* ways to make a hard 15 than a soft 6.

Let’s take the optimal action’s differential value and multiply it by the probability for that circumstance to compute a probability-weighted **true**

value for enacting that optimal action. These true values are shown in the figure below for each scenario, each labeled with the simulation-derived optimal action.



	2		4		6		8		10	
18	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand
	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Surrender
16	Stand	Stand	Stand	Stand	Stand	Hit	Hit	Surrender	Surrender	Surrender
	Stand	Stand	Stand	Stand	Stand	Hit	Hit	Hit	Surrender	Surrender
14	Stand	Stand	Stand	Stand	Stand	Hit	Hit	Hit	Hit	Hit
	Stand	Stand	Stand	Stand	Stand	Hit	Hit	Hit	Hit	Hit
12	Stand	Stand	Stand	Stand	Stand	Hit	Hit	Hit	Hit	Hit
	Double	Double	Double	Double	Double	Double	Double	Double	Double	Double
10	Double	Double	Double	Double	Double	Double	Double	Double	Hit	Hit
	Double	Double	Double	Double	Double	Hit	Hit	Hit	Hit	Hit
8	Hit	Hit	Hit	Double	Double	Hit	Hit	Hit	Hit	Hit
	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit
6	Hit	Hit	Stand	Stand	Stand	Hit	Hit	Hit	Hit	Hit
	Hit	Hit	Stand	Stand	Stand	Hit	Hit	Hit	Hit	Hit

	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand
8	Stand	Stand	Stand	Stand	Double	Stand	Stand	Stand	Stand	Stand
	Stand	Double	Double	Double	Double	Stand	Stand	Hit	Hit	Hit
6	Double	Double	Double	Double	Double	Hit	Hit	Hit	Hit	Hit
	Double	Double	Double	Double	Double	Hit	Hit	Hit	Hit	Hit
4	Hit	Double	Double	Double	Double	Hit	Hit	Hit	Hit	Hit
	Hit	Double	Double	Double	Double	Hit	Hit	Hit	Hit	Hit
2	Hit	Hit	Double	Double	Double	Hit	Hit	Hit	Hit	Hit

	Split	Split	Split	Split	Double	Split	Split	Split	Split	Split
10	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Stand
	Stand	Stand	Stand	Stand	Stand	Stand	Stand	Split	Split	Split
8	Split	Split	Split	Split	Split	Split	Split	Split	Split	Split
	Split	Split	Split	Split	Split	Split	Split	Split	Split	Split
6	Split	Split	Stand	Stand	Stand	Split	Split	Split	Split	Split
	Double	Double	Double	Double	Double	Double	Double	Double	Double	Hit
4	Hit	Hit	Hit	Double	Double	Hit	Hit	Hit	Hit	Hit
	Split	Split	Split	Split	Stand	Split	Split	Split	Split	Split
2	Split	Split	Split	Split	Split	Split	Split	Split	Split	Split

Data generated from 10K hands of H17 game on each combination, natural Blackjacks excluded from count (DDAS/rehit-AA/resplit-AA allowed, max. 3 splits, 6 decks with 4 cut-in)

Amazingly, you can immediately observe that most of the true value of the optimal actions is secluded to a small number of clusters on the tables. As expected, a lot of high value is derived in the hard-total table, especially for situations that call for doubling-down. There are also large swathes of recommended splits in the corresponding table that can be easily aggregated in a simple way (e.g. always split on a card, regardless of dealer). Last, there is a noteworthy cluster of recommended double-downs in the soft-strategy table.

I have tested different digestions of these results into the simplest set of 5 rules I could manage, as summarized in the table below. Many optimal values are missed, and some optimal values are violated in the name of simplicity — but the cumulative effect remains overwhelmingly positive in these few cases.

Rule No.	Action	Player Cards	Dealer Cards
1	Stand	Hard 12 - 16	< 7
2	Double-down	Hard 9 - 11	< 9
3	Split	2-3, 6-8, and A	Any
4	Double-down	Soft 13-18	4 - 6
5	Surrender	Hard 15 - 16	> 9

How do my 5 minimalist rules stack up against the performance of the perfect optimal H17 strategy? Let's take a look!

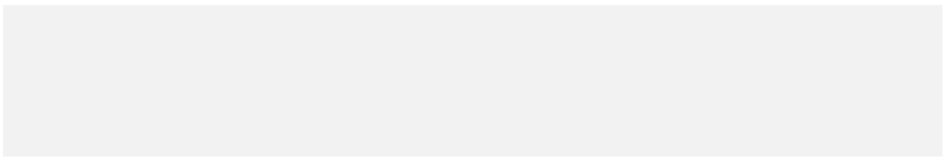
The figure below shows how the inclusion of Rules 1 through 5 increment on the value of optimal strategy relative to the default, dealer-mimetic strategy. Incredibly, **90.6% of the value of optimal strategy** is realized with just these 5 simple rules. What's more, **87.5% of that same value is achieved with only 3 rules!** That's a house edge of 0.76% and 0.93% for 5-rule and 3-rule minimalist strategy, respectively, to be compared with a 0.22% house edge from full optimal strategy.

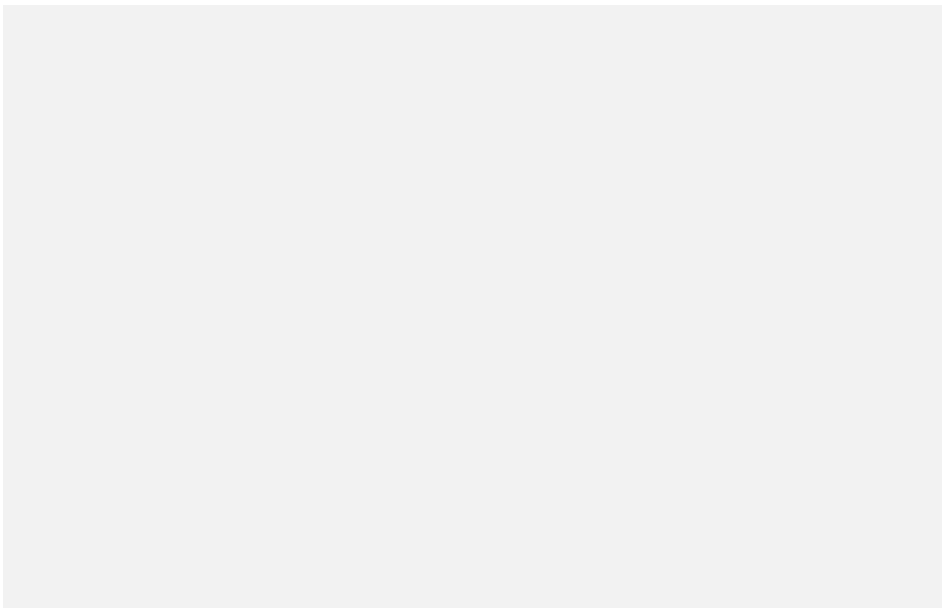
That certainly doesn't say a lot for the other 20-some rules you'd have to memorize for perfect optimal strategy. The implication is immense! If you're a casual Blackjack player looking to do *really well* with minimal effort, look no further.

Data generated from 10MM hands of S17 game with batches of 100K (DDAS/rehit-AA/resplit-AA allowed, max. 3 splits, 6 decks with 4 cut-in, 3/2 Blackjack)

Where does the rest of the value go? Is there any substantial difference in the expected outcomes of our minimalist strategy relative to perfect play?

The answer is simply *no*. The distribution of outcomes for a simulation of 1MM hands is shown in the bar graph below. There are slight improvements to each of the positive outcomes, and decreases for the negative ones, for perfect play. Yet, by and large, we are simply *generalizing* the perfect strategy into a subset of simpler rules for a very similar outcome.





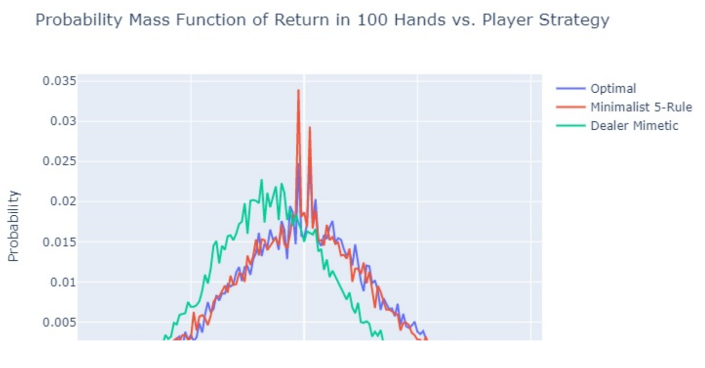
Note: Vertical hash separates outcomes with Double-down (above) from those without (below)

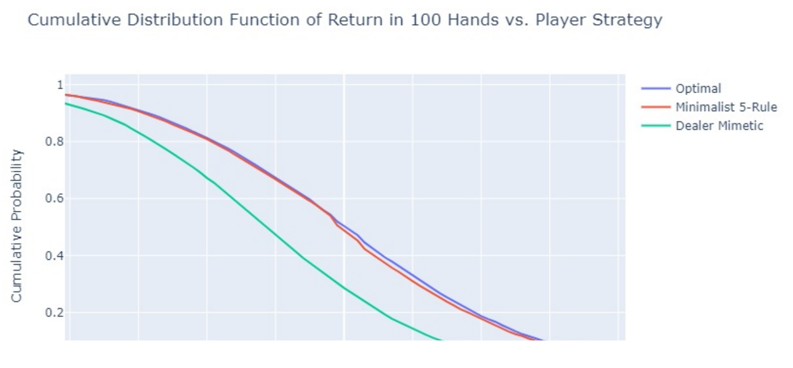
Astute readers will note that the 0.22% house edge is more than 3 times improved on the 0.76% house edge of the 5-rule strategy, so that a perfect player would be expected to last 3 times as long before emptying their chip stack. For very serious Blackjack players who are betting on thousands upon thousands of hands, there is therefore no real exception to the full optimal strategy. That being said, most *serious* Blackjack players who expect to win (not just lose more slowly) rely on card counting anyway.

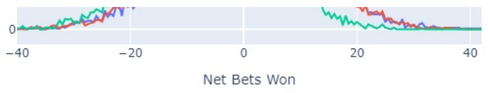
Naturally, frequent players also aren't inclined to cut corners; but this 5-rule minimalist strategy opens the door for amateur card counting techniques to be implemented for a positive player edge that can be attained without much more effort! That is to say, these rules can bring even a casual player to the doorstep of statistically favorable odds in a game that regularly rakes in millions from the less informed!

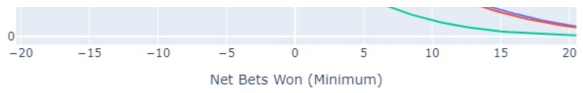
If it's not your cup of tea to sit at a Blackjack table for dozens of hours on end to chase a slight edge before getting flagged for card counting, even better. Take a look at my distribution analysis of the 5-Rule Strategy next to perfect optimal play.

To generate these results, I've simulated 1MM hands and divided this into 100-hand runs (2 hours at a typical dealer rate of 50 hands per hour) as per the *batch means* approach. Each step along the probability mass function (PMF) represents a discrete outcome from one of these 100-hands.









Data generated from 1MM hands of H17 game with batches of 100 (DDAS/rehit-AA/resplit-AA allowed, max. 3 splits, 6 decks with 4 cut-in, 3/2 Blackjack)

The PMF of the minimalist strategy is almost indistinguishable from that of perfect play, save for two significantly higher probabilities at the outcomes of +1 and -1 bet (standard single win/loss). That is to say, the probability of gaining or losing more than 1 net bet is nearly identical for both strategies! Integrating this up to get the empirical CDF, we can also examine the relative probabilities of attaining *at least* N net bets won. If we seek to walk away from our two hour gamble with at least 5 bets won, our probability is 31.0% with the minimalist strategy and 33.0% with the perfect strategy. For 10 bets won, that's 17.7% and 18.7%.

In either case (a net gain of 5–10 hands won in a 2 hour night of Blackjack), the cumulative probability of these short-run gains with perfect play is ***only about 6% higher*** than with the 5-rule minimalist strategy.

So yes, the long-term gains with perfect strategy are *relatively* substantial — but in the short run, this small *absolute* improvement to house edges is nearly inconsequential.

For my money, and my time, I'll stick to the 5 rules above the 24! If I'm ever interested in closing the rest of that house edge, I'd surely just spend another hour or two learning basic HiLo card counting rather than spend dozens of hours training on perfect play for marginally higher returns.

I hope that you'll come to the same conclusion, or at least be satisfied by what we've learned. At any rate, thanks for taking the journey with me!

· · ·

Ode to Streamlit

I would be mistaken to not address the amazing Python library *Streamlit* which I've used here, as many times before, to translate algorithms to user-friendly platforms for interaction and broad deployment. Unlike with alternative packages that demand a proficiency in HTML and networking, *Streamlit* makes all of this possible in just another 100 or so lines of code directly in a short Python script. Check it out at streamlit.io!

Free hosting on compatible sites including Heroku make it possible to bring these projects to life — to share your passion for data science even with your non-programming friends and colleagues!

If you haven't already, pay a visit to the Streamlit app that can be used to generate all of the results I've presented here. If you can use this tool to improve upon what I've learned, I'd love to hear it!

Interactive Blackjack Strategy Simulator

Simulation Optimization Programming Data Science Python