**The Stupidest Card Game Ever.**

When I was 9 I came up with the Stupidest Card Game Ever. It’s very simple and can be played alone, so it’s perfect for a bored kid with nothing to do.

Here’s how it works: shuffle a deck of cards — do it [7 times if you want it to be “truly random”](https://en.wikipedia.org/wiki/Shuffling#Randomization) — and then deal yourself one card at a time. Before you see each card, try to guess the value (2 through Ace, don’t worry about suits). That’s it! Go through the whole deck, and be sure to keep track of your score, which is how many you get right. Obviously, higher is better, and I told myself that any game above 4 was a “good” one, since you could clearly get 4 by guessing the same value over and over again. I think my best score was 8.

So yeah, the game lives up to its name. But it’s surprisingly addictive, especially if you have nothing to do. So around the 6th day of social distancing, when my roommates and I ran out of steam to play Catan or Wizard or dinner-table ping-pong, I introduced them to the Stupidest Card Game Ever. We first tried to guess cards randomly. We got 5 right once, but also got 2, and then 3. Embarrassing.

We turned to a new strategy: counting cards to determine which ones are more likely to show up. It would be nice to be able to keep track of all 13 values, but that was too hard, so we just focused on the first five values. We would start out by guessing 2s, and each time we saw any card valued 2-6, we would mentally take note that we’d seen it. After correctly guessing a 2, we’d change our guess to whichever card we’d seen the fewest number of times. Ties would be resolved arbitrarily — in our case, we always chose the card with the lower value.

[insert table showing your mental table]

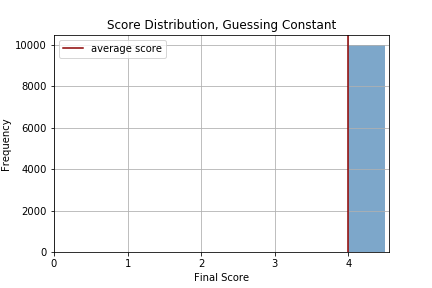
We paired up to be able to keep track of more cards. Each of us would remember 5 cards each, so we could cover the 2s through the Jacks, and we ended up scoring 10. That felt really good. But how good of a score actually was it?

My friends and I had noticed that the shuffle of the deck really affected our score. Sometimes, just by random chance, we happened to guess right early on in the game, and ended up doing well. Other times, it felt like we were spinning our wheels aimlessly, guessing 2s for nearly half the deck before scoring a point. It would be nice to find a way to see how good our strategy was on average, and see if our score of 10 was impressive or not.

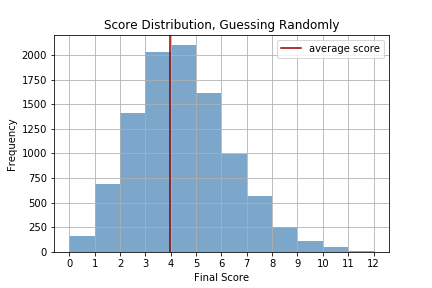
**Visualizing Different Strategies**

Luckily, these types of questions are perfect for modeling with [Monte Carlo simulation](https://en.wikipedia.org/wiki/Monte_Carlo_method). If you could write an algorithm to play the Stupidest Card Game Ever, you could run it tens of thousands of times. The huge number of trials means that the advantage or disadvantage you get from a specific deck configuration gets smoothed over into a distribution of possible scores. You can think of the average of those scores as what you should expect to score in any given game.

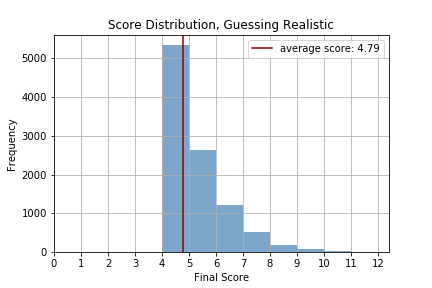
Here’s an example. Remember that first way I thought about playing, where you just guess the same value over and over again? Here’s how the distribution looks. Since you would get 4 *every single time*, it’s not very interesting. Obviously, the average, the minimum, and the maximum are all 4.



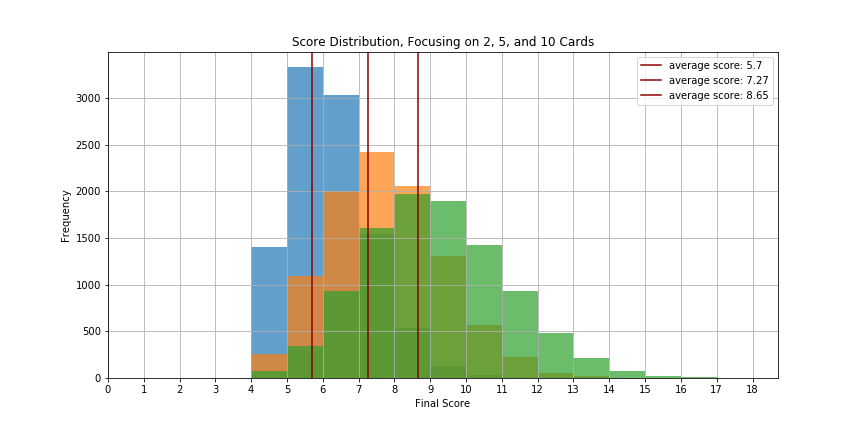
Here’s a slightly more complicated strategy: guessing randomly. Interestingly enough, the average is still roughly 4 (and would approach 4 exactly as you increase the number of trials), but you’ve also got some real variance! If you’re a gambler, this strategy might appeal to you — while you don’t have the certainty of getting 4 in any game, you do have a chance at hitting it big with a score of 10+…or you could get nothing.



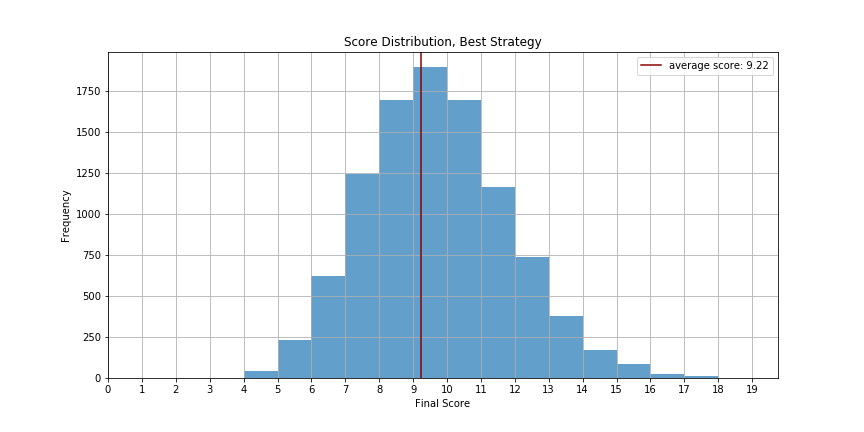
But something was bothering me about the strategy of guessing a constant value every time. At some point, you will have seen (and guessed correctly) all four of your cards, and you still might have some deck left to go through. The constant-guess distribution above assumes that you continue guessing that same value, which doesn’t make sense — any reasonable player would switch it up at that point. So, this is what the score distribution looks like if your strategy is “guess a constant value until you see four of that value, then guess randomly for the rest of the deck.” As you can see, it chops off the left half of the distribution — no matter what, you’re guaranteed to get a score of 4, just like the constant guessing strategy. But you have the potential to score much higher.



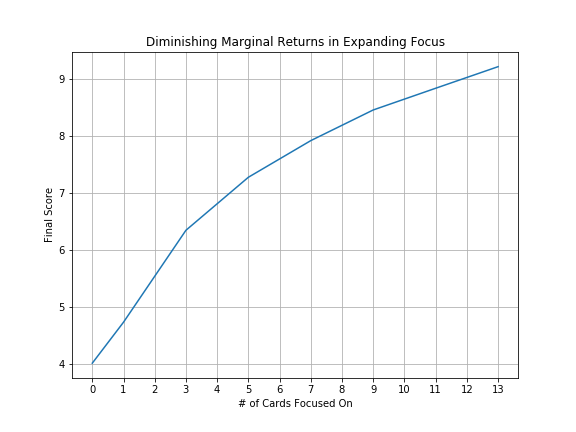
It’s not too hard to model the strategy that we ended up using — where we “focused” on a set of cards, choosing the least-seen card in that set until we had seen all of those cards, and then guessing randomly among the rest. Below are the histograms that arise when focusing on 2 (blue), 5 (orange), and 10 (green) cards.



So what’s the best strategy to win the Stupidest Card Game Ever? That would be to “focus” on all 13 of the values, keeping track of how many times you have seen each value, and continually guessing the card you’ve seen the least, resolving ties arbitrarily. Below is the histogram that arises from this strategy: the average is about 9.2 correct cards, but there is a pretty significant variance. And if you run the simulation long enough, you’re likely to get at least one score of 20.

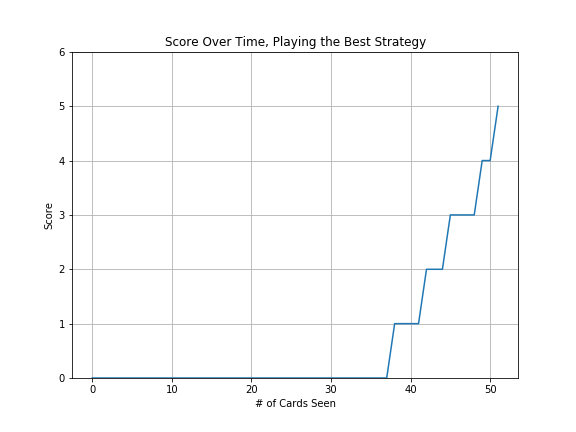
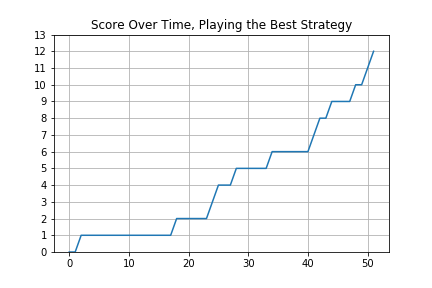


The strategies I’ve thought about have really just been a specific instance of the following general rule: Focus on N cards until you know you’ve seen all of them, and then guess randomly. If N = 0, then this strategy is equivalent to guessing randomly the whole time, and you have an expected score of 4.0. If N = 1, this is equivalent to guessing a constant card until you’ve seen them all, then guessing randomly. This ups your expected score to about 4.8, almost a whole point higher! You can increase N up until N = 13 (equivalent to the best possible strategy) which gives you an expected score of about 9.2. But as the chart below shows, you get diminishing marginal returns from an increased scope of focus. For humans, this means there is maybe a “sweet spot” — a point at which it would not be worth the possible extra points to justify spending the mental energy to keep track of another card.

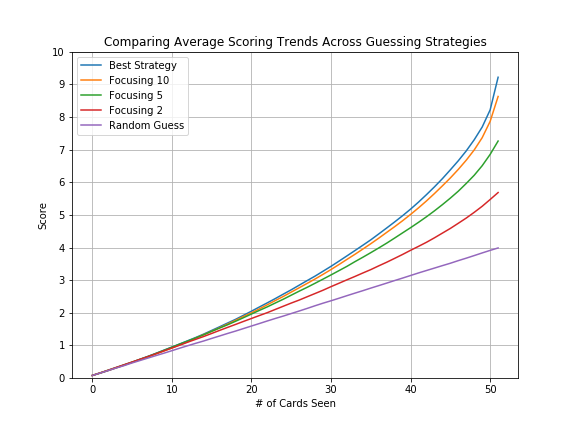
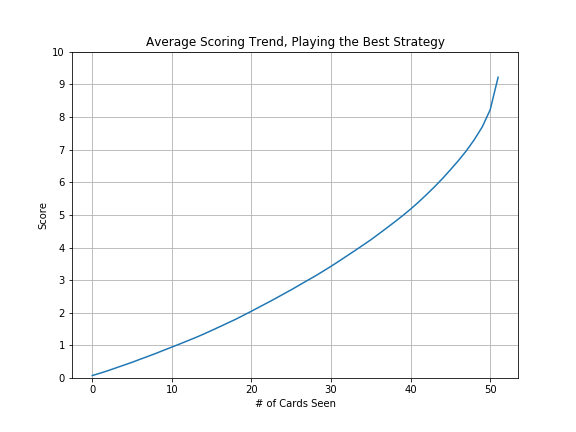


**Scores Over Time**

There are other ways to look at this game. The histograms above illustrate the *final scores* of tens of thousands of games, but it’s also interesting to think about how your score changes throughout a game. As my friends and I played, we noticed that our accumulation of points wasn’t linear. Through the process of elimination, we could be reasonably sure of the last few cards, and much more likely to pick them correctly. For example, the line graphs below show two games that the algorithm played, using the best strategy of focusing on all 13 values. Notice that no matter what, the last card is always guessed correctly — since the computer can keep track of every card, it knows for sure what the last card will be. Also, note that in the game that took a while before a card was guessed right, the final value was only 5, while in the game where the computer guessed right relatively quickly, it ended up scoring 12.



To see the general pattern, we can average out these line graphs over tens of thousands of games played. Below is that curve. It’s decidedly *not* linear. For fun, you can plot different strategies to see how their games progress. Notice that randomly guessing *does* give you a linear score trend-line, which makes sense — it is the only strategy that does not accumulate information as the game progresses, so you will always have the same probability of guessing correctly. However, if you expand your focus and incorporate more information, the nonlinear effect becomes more pronounced.



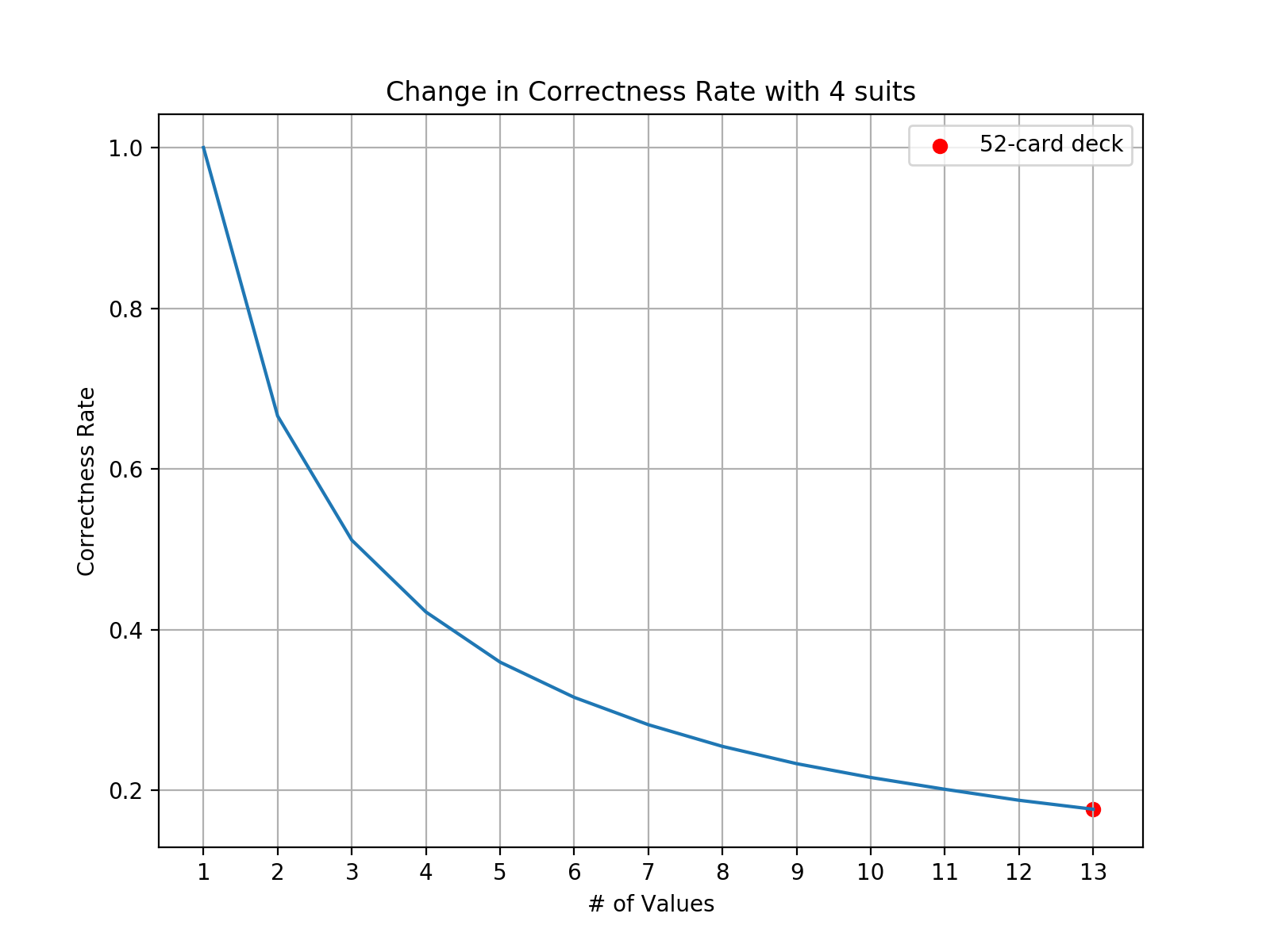
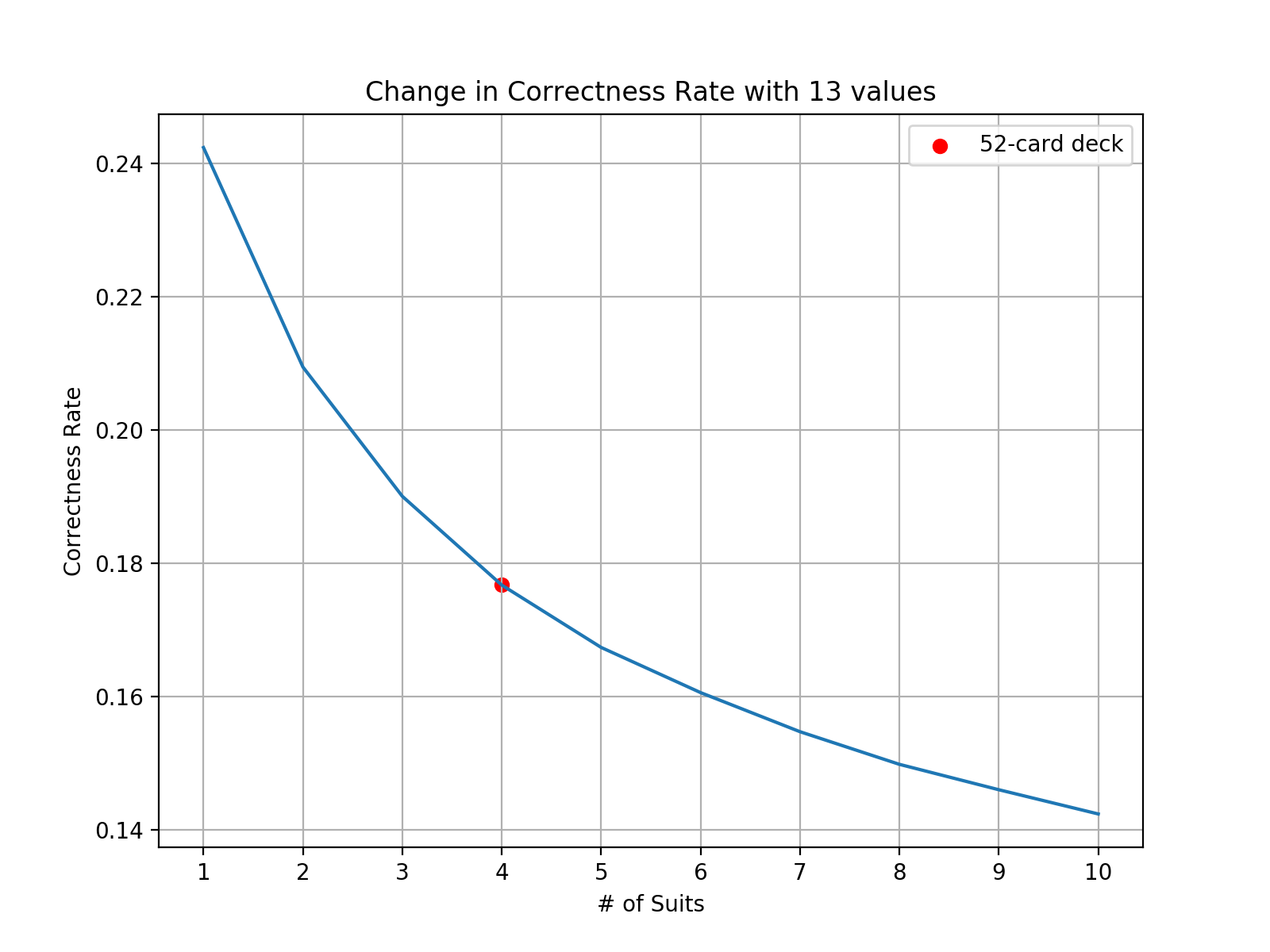
**Getting really weird.**

Ok, turns out you can go even a little bit deeper. Something had been bothering me about this average of 9.2 given the best possible strategy. What makes 9.2 so special? What is it about a 52-card deck with 13 values and 4 suits, that ensures that the best possible strategy gives an expected score of ~9.2? It would be interesting to play the game with a different deck, with a different number of values or suits, and see how that affects the winning score.

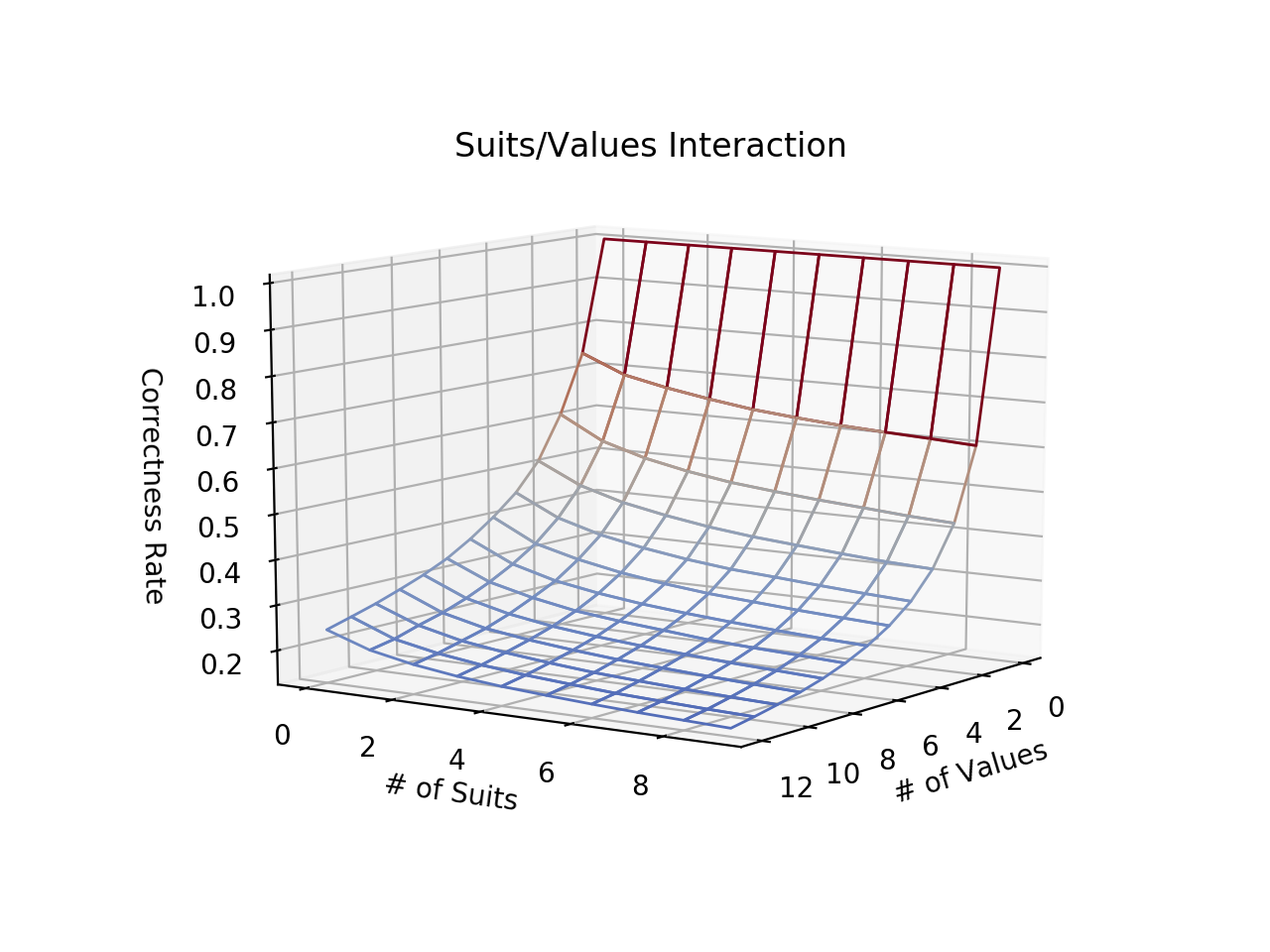
First of all, since we’re now talking about variable deck sizes, it no longer makes sense to talk about score in terms of a raw number, so we can normalize by the deck size. Instead of scoring 9.2, then, our best strategy in the original game gives us a 9.2/52 ~= 17.7% correctness rate.

My first question: is this rate solely based on the number of values you need to keep track of, from 2 to Ace? To check this, I ran the model on a hypothetical 26-card deck with the same 13 values but only 2 suits. The result: an average of 5.4/26 = ~20% correctness rate — noticeably higher than our original value! This makes sense for a couple reasons: first, the fact that there are fewer suits means it is “easier” to know when you’ve seen all of one value of card. Once you see your second 2, for example, you can write them off for the remainder of the deck. Having to wait to see four 2s could make your probability of scoring lower. Another constant in the game is that your last card will always be guessed correctly, so the smaller the deck size, the more of an effect that will have on your correctness rate.

It seems like both the number of values and the number of suits in a deck interact to determine the correctness rate we see from playing this Stupid game. Below are two ways to see this interaction — the plot on the left shows the correctness rates for a deck with 13 values with varying numbers of suits per value (from 1 – 10). The plot on the right shows the rates for decks with 4 suits, with varying numbers of values (from 1 – 13). Both relationships are nonlinear, and both show that as you increase the number of suits or values, your correctness rate seems to decay (exponentially?). However, while varying the values from 1 – 13 swings your possible correctness rate wildly (from 100% down to 17.7%), the suits have a much smaller effect, bringing down your possible rate from around 25% to around 15%.



Since this is really a two-dimensional input space, we might as well make a 3D graph to see the full picture. Here it is!



That’s about as far as I could go into analyzing this silly little game. For what it’s worth, scoring 10 points while keeping track is above average, so I’m pretty pleased.