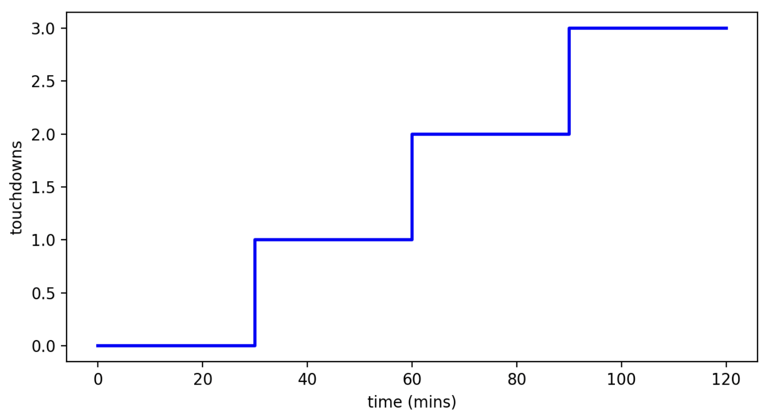
Jeffrey Mu - APMA350 write-up

Let's suppose you have secured a coveted front-row seat at the Superbowl, and you're looking to predict, to some degree, the outcome of the big game.

Specifically, you focus on the touchdowns, since you know these six points are the most significant by far. You're looking to derive some model that can tell you how many points will be scored by your favorite team, the Eagles, because you've put quite a bit of money into this, and your rent for next month won't go to waste.

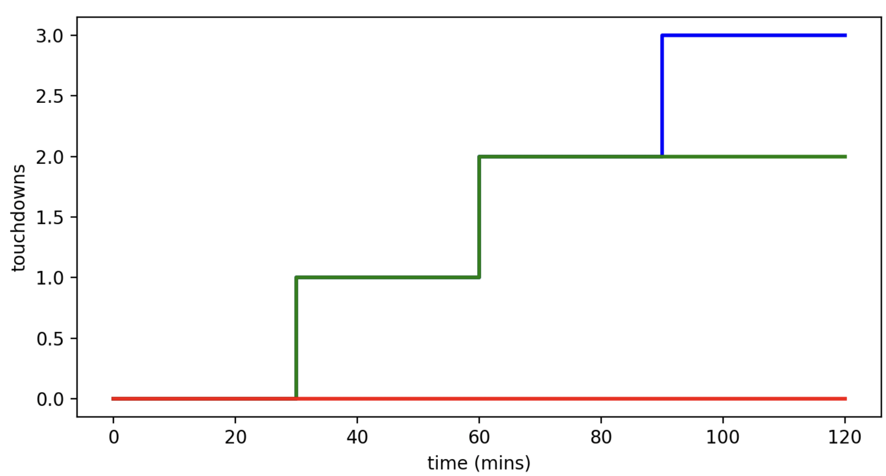
Maybe you've done some research behind the scenes. For the Eagles, across all their previous games, it seems every 30 minutes, the probability of the team scoring a goal is P(goal_single) = 0.6. Obviously, each goal should be independent of each other.

So you're at the beginning of the game, having your model in mind, and one prediction of the trajectory could be something very consistent with the step function, specifically that every 30 minutes, a touchdown is scored. Now, what's the probability that this is indeed the correct representation of how the game goes?



Well, since it's a 60% chance that a goal is scored within that 30-min interval, P(3_goals) = 0.6x0.6x0.6 = 21.6%.

Similarly, we could also predict that up until the 90-min mark, another possible outcome of the game could be identical to the previous one, but instead, between 90-120 minutes, no goal is scored (green), which would be



P(2_goals) = 0.6x0.6x0.4 = 14.4%. Alternatively, we could predict P(0_goals) = 0.4x0.4x0.4 = 6.4% (red).

Okay, these models are cool, but how accurate are they? How ecological is it that for this football team, the goals are made exactly within each 30-min duration, no more, no less, and this recurs throughout the entire game? What if we wanted to analyze overall scoring trajectory distributions with the context that time continuously evolves, and scoring is probabilistic, as is in the real world? How would we go about doing this?

To step away from deterministic models, we use **stochastic differential equations** (SDEs), which are really cool equations that can be used to model anything from the stock market, to the weather forecast, to human behavior. They show up everywhere, well, that can be anywhere that randomness plays a role.

To be more specific, let $X(t)$ represent the total number of touchdowns scored by the Eagles up to time t. Suppose that we follow the rule that every 30 minutes, a touchdown is scored. Since touchdowns happen at random times, and let's assume at an average rate, and each event causes a jump in the score, we can capture random fluctuations more smoothly with a scoring process as a stochastic ODE of the form:

$$dX(t) = \lambda\, dt + \sigma\, dW(t)$$

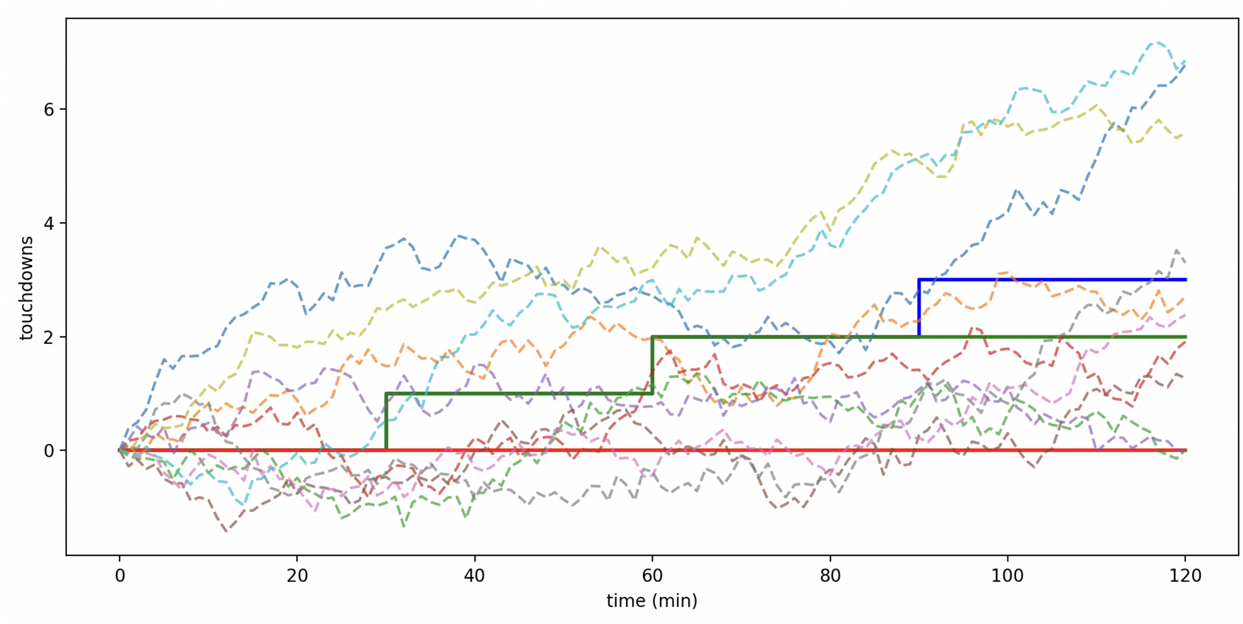
where:

- λ is the average scoring rate (0.6 touchdowns per 30 mins => 1.2 per hour)
- σ captures the variability in the scoring behavior
- $dW(t)$ is the Wiener process, which is used to model the random noise, where $dW(t)$ is a random variable, and following a normal Gaussian distribution with a mean of 0 and a variance of $d(t)$

This equation says that the change in score over time is mostly governed by the average rate of scoring λ, but also perturbed by randomness $\sigma\, dW(t)$.

Interpreting this, maybe once in a while, a field goal, which is half of the touchdown, is scored, so our step function jumps up by 0.5. Or maybe during one 30-min interval, no touchdowns are scored, or two touchdowns are scored, or 5. σ controls for this element of noise, in which unless the game is scripted, there is no way to predict independent events from each other and exactly when they would occur in-game.

More specifically, if we let this function progress on as t continuously evolves to infinity, we would have something like this for a few simulated SDE trajectories:



As someone studying cognitive science, models like these are super interesting because they can be used to show how humans experience uncertainty and generate predictions. In fields like decision-making, reinforcement learning, neural modeling, SDEs help describe how internal states evolve with continuous and often noisy input, and as experimenting with human vision and creativity will tell you, noise is ubiquitous in high-dimensional human data. It's a way to measure responses to stimuli that I might try to apply in the future.

The cool thing is that unlike our previous fixed step-function, the SDE gives us a whole distribution of possible score trajectories, not just one. Each realization of $X(t)$ would look a little different, some with a noise curve, some flat stretches and some jumps. Over a long period of time, it gives us that behavior and randomness that governs a more ecological football setting.

So as we know it now, the Super Bowl is in fact scripted, and you will probably lose most if not all your money. If it weren't, though, the stochastic ode would probably be your next best bet.

References:

Repo: https://github.com/jeffreymu1/SDEs

- Herzog, F. Stochastic Differential Equations
  https://ethz.ch/content/dam/ethz/special-interest/mavt/dynamic-systems-n-control/idsc-dam/Lectures/Stochastic-Systems/SDE.pdf

- Ipython
  https://ipython-books.github.io/134-simulating-a-stochastic-differential-equation/

- Jahnke, Tobias. A short introduction to stochastic differential equations
  https://www.math.kit.edu/ianm3/lehre/numamathfin2023w/media/intro-to-sdes.pdf

- MIT OCW
  https://ocw.mit.edu/courses/18-s096-topics-in-mathematics-with-applications-in-finance-fall-2013/a671d0d2abe626c371fd5850ad670397_MIT18_S096F13_lecnote21.pdf

- Science Direct
  https://www.sciencedirect.com/topics/mathematics/stochastic-differential-equation

- This Quora page:
  https://math.stackexchange.com/questions/523724/why-do-people-write-stochastic-differential-equations-in-differential-form

- Zobitz, J. Exploring Modeling with Data and Differential Equations Using R (I didn't use R)
  https://jmzobitz.github.io/ModelingWithR/sdes-25.html