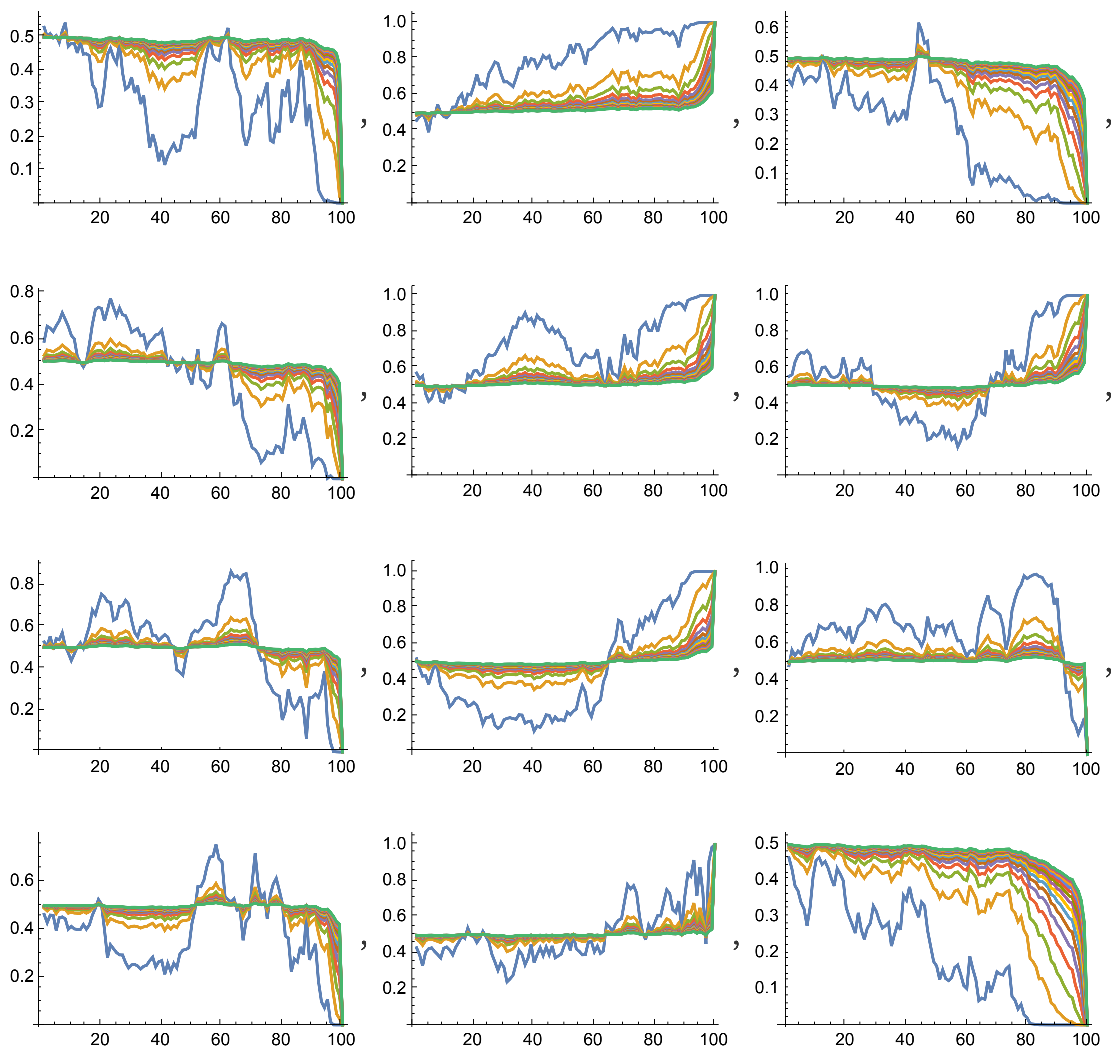
How do you forecast an election?

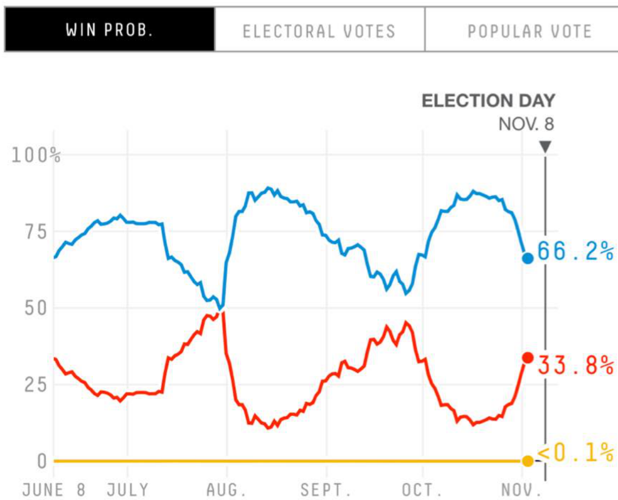
Nassim Nicholas Taleb

DRAFT - CANNOT BE CITED YET. I need to make the notations uniform across the two parts.

A Dynamic View of Forecasting



■ Figure 1: A collection of forecasters for the same variable {0,1} over 100 periods. The blue has little uncertainty in his forecast. The most efficient forecaster is half way, closer to the green line

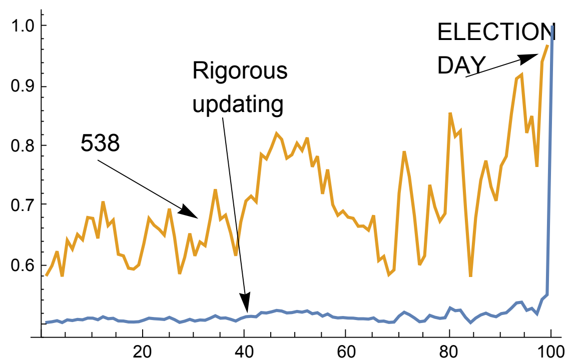


- Figure 2: The defect of 538. They responded to this criticism with ... more ignorance of probability. Yes we have known for >200 years since Laplace's argument that uncertainty and ignorance makes odds remain close to 1/2.

This note is organized as follows. I discuss the option approach than show how it corresponds to de Finetti's approach to minimize the Brier Score as a "proper" score. Note the following:

- The higher the uncertainty, the closer the probability in two-contest need to be at .5
- The higher the uncertainty in the system the more slowly forecast need to update until the final result.

Assume W is a continuous state variable determining the final result.



Some mathematical derivations

Let us start the model from the very basics. Very very basics of stochastic calculus. We have the election estimate *F* a function of a state variable *W,* a Wiener process WLOG. *W* can be an estimate, or some other variable. the estimation error can be integrated into the variance of *W*. *W* has for simple dynamics (arithmetic B M, we can transform later):

$$dW = dt\,\mu + dZ\,\sigma \tag{1}$$

By Ito' s Lemma:

$$dF = \frac{\partial F}{\partial t} dt + \frac{\partial F}{\partial W} dW + \frac{1}{2} \frac{\partial^2 F}{\partial W^2} dW \quad (2)$$

Ito's calculus allows dt^2 and $dt\, dW$ to vanish. The idea of no arbitrage is that a continuously made forecast must itself be a martingale of sorts. Apply the Black Scholes (or a standard no arbitrage) argument; Replacing with (2), and assume μ=0 to simplify WLOG

$$dF - \frac{\partial F}{\partial W} dW = 0 \quad (3)$$

We end up with the partial differential equation:

$$\frac{\partial F}{\partial t} dt = -\frac{1}{2} \frac{\partial^2 F}{\partial W^2} \sigma^2 \quad (4)$$

which is, basically, the heat equation. We have for terminal conditions: $F|_{t\to 0} = \theta[W]$ where θ is the Heaviside Theta function. We can try to solve on Mathematica (by fudging, inverting the backward-forward equation)

```
Eq = D[F[W, t], t] == 1/2 σ^2 D[F[W, t], {W, 2}];
tc = F[W, 0] == HeavisideTheta[W];
sol = DSolve[{Eq, tc }, F[W, t], {W, t}]
```

$$\left\{\left\{F[W, t] \to \frac{1}{2}\left(1 + \text{Erf}\left[\frac{W}{\sqrt{2}\ \sqrt{t}\ \text{Abs}[\sigma]}\right]\right)\right\}\right\}$$

which is the CDF of a the Normal distribution for P≥W. If W is a "poll", we can transform ϕ^{-1}:(-∞,∞) → [0,1] to get it to translate.
E finito!

Connection to De Finetti' s Approach

What makes a good forecaster? As traders we know that the final outcome is just a piece of the pie. Every day's P/L matters. You need to consider the steps in the process. In fact, at some point, you can tell a bad forecaster before the end event, and tell **when** you can pronounce forecaster A better than forecaster B —for no matter the final outcome A will dominate B. In the real world, a forecaster who is also a market maker **can go bankrupt** before final outcome.

The idea of a "proper score" is as follows. It is simply, a method that penalizes you if your distribution of outcomes diverges from the "real" probability distribution.

Also this shows how it is worse to **produce no change in forecast** than keep changing, and how to calibrate changes to volatility.
The math is as follows. Let b_{t_0} be your "price" ∈[0,1] time t_0, your "probability", and $b_{t+\Delta t}$ your price time $t+\Delta t$, etc. Assume elections happen time τ.
Since your forecast is left hanging, you are evaluated at *how little opportunity one can arbitrage you*,

that is buy from you at b_{t_0} and sell at $b_{t_0+\Delta t}$. Hence your quality of forecasting is some norm $\| b_{t_0} - b_{t_0+\Delta t} \|_2$. This relates with the Brier metric which would be $\| b_{t_0} - b_\tau \|_2$, $b_\tau \epsilon \{0,1\}$ being the final result. Note the Brier metric uses Norm L2 (squared deviations) but your P/L is is norm L1 (absolute deviations) but the former is preferable because it is a "proper" score.

$$\mu(t_0, \tau, \Delta t) := \frac{1}{n} \sum_{i=0}^{\frac{\tau - t_0}{\Delta t}} | b_{(i+1)\Delta t + t_0} - b_{i \Delta t + t_0} | \tag{5}$$

$$\mu_2(t_0, \tau, \Delta t) := \frac{1}{n} \sum_{i=0}^{\frac{\tau - t_0}{\Delta t}} (b_{\Delta t (i+1) + t_0} - b_{i \Delta t + t_0})^2 \tag{6}$$

Of course a series of Brier scores

$$\mu_B(t_0, \tau, \Delta t) := \frac{1}{n} \sum_{i=0}^{n} (b_{\Delta t (i+1) + t_0} - b_{n+1})^2, \; n = \frac{\tau - t_0}{\Delta t} \tag{7}$$

The probabilist can see that as $\Delta t \to 0$ we have a nonanticipating Ito integral for the L2 norm.

Next let us see how a dynamic forecaster using Ito's lemma ***minimizes*** the Brier score.

```
In[39]:= DiffBrier[vec_] := 1/Length[vec] Sum[(vec[[i]] - vec[[Length[vec]]])^2, {i, 2, Length[vec]}]

Brier[m_, σ_] :=
 Table[ta = Join[{0}, RandomVariate[NormalDistribution[0, σ], 100]] // Flatten //
     Accumulate;
   ta1 = Table[CDF[NormalDistribution[0, Max[.0001, m σ Sqrt[Length[ta] - i]]],
      ta[[i]]], {i, 1, Length[ta]}];
   DiffBrier[ta1], {i, 1, 2 × 10^5}] // Mean
```

We can see the Brier is flat in σ as both scale equally.

```
In[21]:= Table[Brier[1, σ], {σ, 1, 5, 1}]

In[41]:= tt1 = Table[{m, Brier[m, 1]}, {m, 1, 5, 1}]

Out[43]= {{1, 0.15599}, {2, 0.166106}, {3, 0.178219}, {4, 0.187488}, {5, 0.194707}}

In[29]:= tt2 = Table[{m, Brier[m, 1]}, {m, .1, 2, .1}]

Out[54]= {{0.1, 0.215846}, {0.2, 0.195365}, {0.3, 0.182563}, {0.4, 0.172056}, {0.5, 0.168916},
  {0.6, 0.162945}, {0.7, 0.16113}, {0.8, 0.159371}, {0.9, 0.157854}, {1., 0.157015},
  {1.1, 0.157217}, {1.2, 0.157812}, {1.3, 0.158104}, {1.4, 0.15875}, {1.5, 0.160005},
  {1.6, 0.161152}, {1.7, 0.162826}, {1.8, 0.163945}, {1.9, 0.164491}, {2., 0.166177}}
```

In[57]:=
```
ListPlot[Join[tt1, tt2], PlotStyle → Red, AxesLabel → {σ, Score}]
```

Out[57]=