to appear in Journal of Reproducible Statistics

The Squealer: Sensification of model exploration and model misfit

Andrew Gelman[*], Andrew H. Jaffe[†], Eliot Carlson[‡], Philip Greengard[§], and Jeremy Magland[‡]

20 Nov 2026

Abstract

We introduce a method for visual and auditory feedback when exploring the fit of a model to data. Starting with a best-fit curve fit to data, the user can drag the curve to a new position and the computer will emit a squeal, becoming more unpleasant as the discrepancy between curve and data increases. We demonstrate with four examples: a two-parameter curve fit to golf putting data, a four-parameter curve fit to dilution assays, a fit to cosmological data sensitive to the parameters of the Big Bang model, and a nonparametric Gaussian process fit to temperature readings. In this form of dynamic exploratory data analysis, the user employs domain knowledge to consider directions for exploration and to interpret the results.

1. A method for dynamic exploration of fitted models

Visualization is important, not just for data exploration but also for statistical modeling (Gelman 2004; Wickham 2006; Wickham, Cook, and Hofmann 2015). By plotting the estimated model alongside the data, we can visualize the model and also see places where it does not fit.

When fitting a model, often we are in a situation in which the best-fit curve systematically misses some of the data, and we would like to "grab" the curve and drag it to a new place. Presumably that would create new problems—if we are starting at the best-fit curve, any alternative would represent a greater aggregate discrepancy—but we would like to see where these problems are, and which particular data points are causing the problem.

1.1. Motivation and formulation of the problem

We formulate the simplest version of the problem in terms of a Bayesian model, $p(\theta|y) \propto p(\theta)\prod_{i=1}^{N} p(y_i|x_i,\theta)$, with the curve of interest being $g(x|\theta) = \mathrm{E}(y|x,\theta)$, which we assume can be computed analytically. The data x_i and y_i are scalars and θ is a parameter vector. More general curve-fitting posteriors can also be accommodated in the formalism.

We assume we can fit the model and estimate θ, which implies an estimate of the curve, $g(x)$, which can be overlaid on a scatterplot of (x, y) and represent a starting point for our procedure. At this point, the fit of the curve could be explored by moving the parameter vector θ, which would in turn move $g(x)$.

Our key idea is to do this the other way around: instead of changing θ and seeing what this does to g, we move g and then adjust θ accordingly. Altering g in a certain direction will not necessarily correspond to a unique choice of θ, so we will do the adjustment of θ in the context of the model to provide the best possible fit (more specifically, maximize the posterior density, $p(\theta|y)$) in a way that is consistent with $g(x|\theta)$ being perturbed in the desired direction.

Figure 1 demonstrates the idea in a setting where the data were generated from the class of models being fit. As an example we are using a four-parameter logistic model fit to laboratory assays, as discussed further in Section 2.2. The user should be able to explore the fit by moving the curve on the space of (x, y) and seeing how this translates to changes in the parameters θ and

[*]Department of Statistics and Department of Political Science, Columbia University, New York.
[†]Department of Physics, Blackett Laboratory, Imperial College, London.
[‡]Center for Computational Mathematics, Flatiron Institute, New York.
[§]Center for Computational Neuroscience, Flatiron Institute, New York.

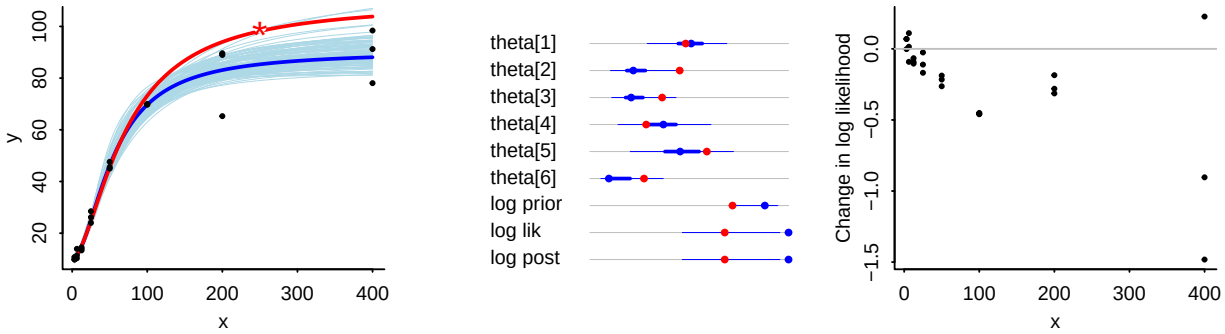


Figure 1: *The basic Squealer.* Left: *scatterplot of data and fitted model (dark blue curve representing the point estimate $\hat{g}$ and light blue curves representing posterior simulations g^*), a pseudo-data point (x^*, y^*) in red, and the new curve g^* in red.* Center: *dashboard showing, for each parameter θ_k in the model, the point estimate $\hat{\theta}_k$ as a blue dot, the posterior density from the simulations θ^s in blue, and the perturbed value θ^*_k in red. The dashboard also shows the corresponding quantities for the log prior, likelihood, and posteriors.* Right: *Relative discrepancy, $\log p(y_i|x_i, \theta^*) - \log p(y_i|x_i, \hat{\theta})$, for each of the n data points plotted vs. x. In the dynamic version of the Squealer, the pseudo-data point would be specified by the user, the change in total fit, $\log p(\theta^*|x, y) - \log p(\hat{\theta}|x, y)$, would be conveyed by sound, and the relative discrepancies of the individual data points would be displayed as different intensities of color of the points on the graph.*

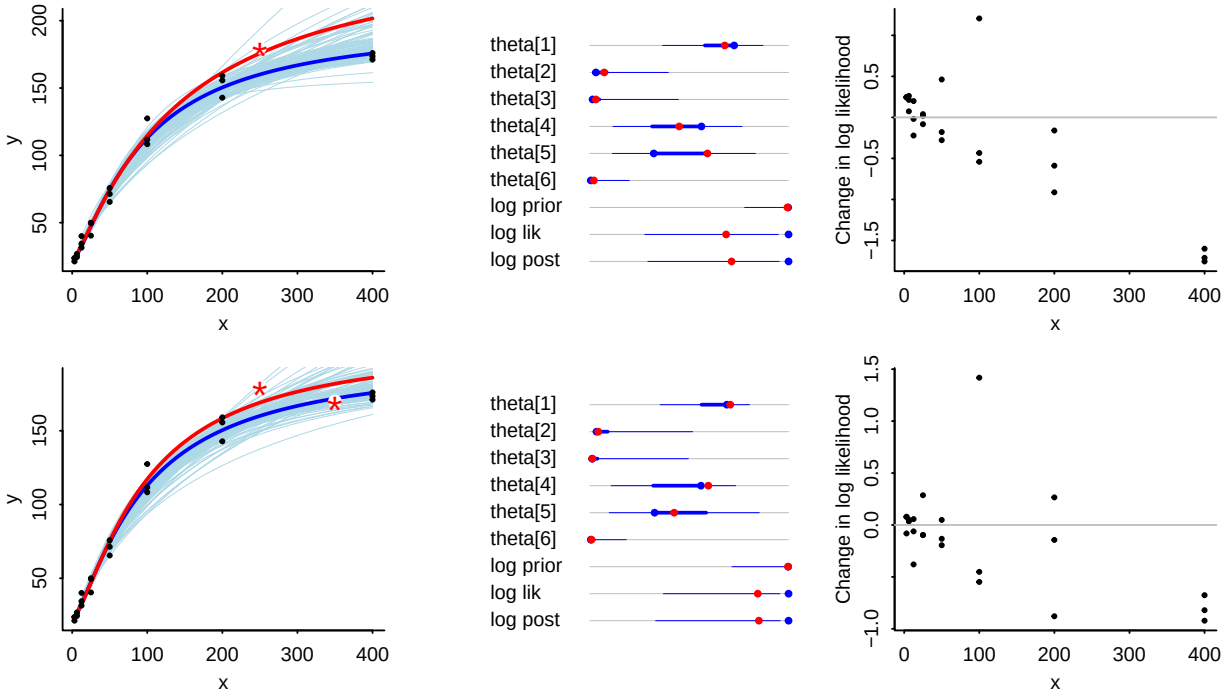


Figure 2: *Example of the Squealer for data that are not consistent with the model. In this case there is no way to get the curve close to the data: pulling up the curve to improve the fit for the point at $x = 200$ degrades the fit elsewhere. The above displays show two attempts, first adding one pseudo-data point and then adding another. In every case, pulling toward the pseudo-data decreases the log posterior density (the lowest line on the dashboard in the middle of each row of three graphs), which makes sense given that we are starting from the posterior mode of the parameters.*

what this does to the fit of the model to the individual data points and in total. A challenge here is that, if θ is high-dimensional, we do not want to require the user to specify all its parameters; rather, we want to be able to move the curve $g(x|\theta)$ in some general direction and let the parameter vector θ follow in a way that is most consistent with the data.

Figure 2 shows an example where the data were generated from a model outside the fitted family, so that the best-fit curve (in blue on the left plots) has systematic discrepancies with the data. The two rows of plots show two attempts to get a better fit by dragging the curve toward the pseudo-data points shown in red on the left plots.

If we knew that the assumptions underlying our model were correct, there would be no reason to do this, as we could just summarize uncertainty using the Bayes posterior distribution. Realistically, though, our models will be wrong, and we are developing a set of tools for a new and dynamic way to understand misfit to data.

The present paper focuses on the statistical method and computational concerns, not on the dynamic graphics or the sonic and visual implementation.

1.2. Perturbing the posterior mode

The simplest version of the Squealer works with the estimate $g(x|\hat{\theta})$, where $\hat{\theta}$ is the mode of the posterior density of the fitted model, $p(\theta|y)$. The user can drag the curve in some direction by pulling it toward to a newly specified pseudo-data point, (x^*, y^*). We do this by multiplying the posterior density by a factor, normal$(y^*|g(x^*|\theta), \sigma^*)$, where σ^* is set to a value small enough that the new point will exert a strong enough "gravitational force" but not so small as to induce computational instability. A reasonable first choice is half the posterior standard deviation of $g(x^*)$, that is, we set

$$\sigma^* = 0.5\,\mathrm{sd}(g(x^*)|y) = 0.5\left(\int (g(x^*|\theta) - \bar{g}(x^*))^2\, p(\theta|y)\, d\theta\right)^{1/2}, \quad \bar{g}(x^*) = \int g(x^*|\theta)\, p(\theta|y)\, d\theta, \quad (1)$$

which should be easy enough to calculate, exactly or approximately, given that the model has already been fit. We then compute the posterior mode, θ^*, for this augmented model. Using the normal distribution as a baseline, we should expect this to shift the curve approximately 80% (that is, $\frac{1^2}{1^2+0.5^2}$) toward the pseudo-data point; the amount of this shift would be different for other models.

In the computation, we use the existing estimate, $\hat{\theta}$, as the initial value for the optimization. There could be computational challenge, however, if the pseudo-data point is far from the fitted curve. One approach is to check if $|y^* - \mathrm{E}(g(x^*|y))| > \mathrm{sd}(g(x^*|y))$ and, if so, shift y^* gradually, moving it by $\mathrm{sd}(g(x^*|y)$ at a time until we eventually reach the specified value.

In assuming a normal distribution for this gravitational force, we are *not* assuming that the likelihood for the data y_i necessarily follow a normal distribution. The pseudo-data point does not have to take the mathematical form of an additional measurement from the model, $p(y|x, \theta)$; rather it is entirely a mathematical and computational tool for gradually moving the curve from $\hat{g}$ to a new estimate, $g^* = g(x|\theta^*)$, obtained by computing the mode of the augmented posterior distribution of θ. For example, if $p(y|x, \theta)$ is a discrete-data model such as an ordered logistic regression of a response with five categories, it would still make sense to choose a continuous value for y^*, as we are using it to shift the continuous curve g, not to represent a possible data point.

For a complicated model it can make sense to move the curve in more elaborate ways, by pulling it toward multiple pseudo-data points, (x_i^*, y_i^*), $i = 1, \ldots, N^*$. The same logic holds; we merely multiply the posterior density by $\prod_{i=1}^{n^*}$ normal$(y_i^*|g(x_i^*|\theta), \sigma_i^*)$, setting each σ_i^* using some rule such as (1), and then obtain a new estimate, θ^*, and a corresponding new curve, $g^* = g(x|\theta^*)$.

In either case, the new curve g^* represents the best fit prediction given the model and data, under the soft constraint that the curve passes near the new point or points, (x^*, y^*). When a model is complicated and has many parameters, we do not want to be in the position where we have to specify the entire curve; rather, we want to be able to gently drag it and then let the data dictate the best fit given our soft constraints. Figures 1 and 2 show examples.

On the computer, the user should be able to perform this operation in one of two ways, either by clicking on the location of the pseudo-data points or by clicking on a point or points on the fitted curve $\hat{g}$ and then using the mouse or joystick to move these points to new places, which become the pseudo-data points (x^*, y^*). In either case, this would be performed using dynamic graphics on the plot of data and fitted curve, and then in the background the computer would re-fit the model and display the updated curve.

Depending on the speed of computation, there are different ways that the updated curve could be estimated and displayed. If the fit is fast enough, the computer could re-fit the model on the fly as the curve is being dragged so that the user could see the updated curve moving in real time. If real-time updating is impossible, it could be possible to compute the updated model using an annealing-like procedure in which the gravitational scale, σ^*, starts at a high value and then gradually decreases to zero, and the fitting algorithm can provide real-time updates.

1.3. Perturbing a posterior mean or other point estimate

The mode is not always a good summary of the posterior distribution: the best-fit curve will always overfit to some extent, and even in a simple multivariate normal posterior distribution, the mode will not be in the typical set as the dimensionality increases (Gelman 2020). And, even in low dimensions, the posterior mode can have problems. For a hierarchical model with weak priors the joint mode occurs at the degenerate point at which all the local parameters are equal to each other and the group-level variance is zero.

For these reasons, we often choose posterior summaries other than the mode. Two alternatives are the mean and median, either of which can be computed using posterior simulations θ^s, $s = 1, \dots, S$, or from a variational approximation to $p(\theta|y)$. For the Squealer, we are interested in $g(x|\theta)$, not θ itself, so if we have an estimate $\hat{\theta}$ (which could be the posterior mode, mean, median, or a summary of a variational inference), we can use $\hat{g}$ defined by $\hat{g}(x) = g(x|\hat{\theta})$.

Alternatively we could take the posterior expectation of the curve itself, so that $\hat{g}(x) = \mathrm{E}(g(x)) = \int g(x|\theta)p(\theta|y)$. This is appealing from a theoretical perspective given that we have already defined $g(x|\theta)$ as $\mathrm{E}(y|x, \theta)$, so that this new $\hat{g}$ is simply the unconditional posterior expectation. There is a computational challenge here, though, in that this averaging would need to be evaluated at a grid of values of x, unlike the other approach in which the point estimate $\hat{\theta}$ only needs to be computed once and then it can be plugged into the formula for $g(x|\hat{\theta})$.

Whatever choice of distributional summary we use, the plan is to first compute it from the posterior distribution, $p(\theta|y)$, and then on the augmented posterior distribution, $p(\theta^*|y) \propto p(\theta|y)\mathrm{normal}(y^*|g(x|\theta^*), \sigma^*)$, using the inference from $p(\theta|y)$ to initialize the computation.

If the original model has been fit using posterior simulations, one might hope to be able to shift the curve by importance reweighting of the simulation draws using the normal density corresponding to the pseudo-data. It would then be fast to compute the mean of θ under the augmented posterior distribution or to compute $\mathrm{E}(g(x|\theta))$ in parallel at a grid of values of x. Unfortunately we often will be moving the curve to values that are barely supported by the original model—that is one of the reasons for developing this method in the first place, to understand why a certain alternative fit does not work for the data at hand—, in which case simple importance weighting would not work. It could make sense to use some sort of particle filtering algorithm, but here we will assume

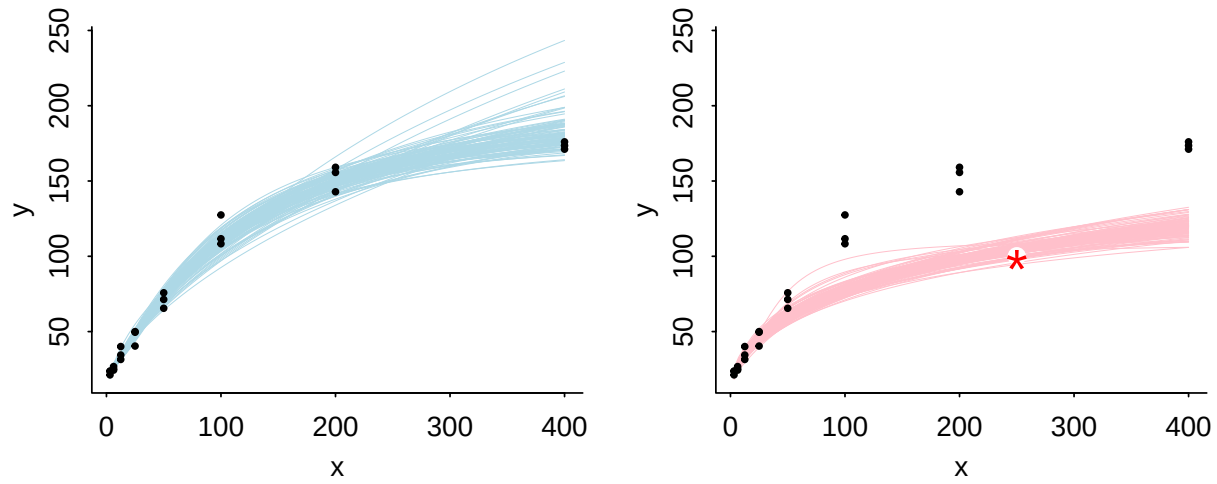


Figure 3: *Challenge of shifting the posterior distribution by pulling it toward a pseudo-data point.* Left graph: *data and fitted curves $g(x|\theta^s)$ based on 100 random draws θ^s from the posterior distribution, $p(\theta|y)$, for the data and model shown in Figure 1.* Right graph: *data and pseudo-data point, along with fitted curves based on 100 random draws from the augmented posterior. The curves on the right are constricted by the "local gravity" of the pseudo-data point.*

that we will just fit the augmented model directly.

When perturbing $g(\hat{\theta})$, where $\hat{\theta}$ is the posterior mode, the log posterior density necessarily decreases. When perturbing a $\hat{g}$ using the posterior mean or median or other summary, there is no guarantee that the log posterior will decrease, so it is not so clear what should be "squealing" to indicate discomfort with the fit. One approach might be to follow a variational framework and define discomfort as the Kullback-Leibler discrepancy of $p(\theta^*|y)$ relative to $p(\theta|y)$. In this case, the point estimate $\hat{g}$ is there for the graphical display but would not be used to summarize total error.

1.4. Perturbing a posterior distribution

For Bayesian applications we are often interested in the entire posterior distribution, and we would like to see what happens if it is shifted in some way. Unfortunately, our "gravitational" approach leads to an artifact by which the posterior distribution of the shifted curve g^* is constricted in the neighborhood of each pseudo-data point. We would like to have a way to shift the entire distribution without pinning it in any particular place. One idea that might seem appealing is to shift the posterior draws of θ by a fixed amount, that is, to define simulations $\theta^{*s} = \theta^s + \theta^* - \hat{\theta}$, but that gives poor results in examples such as this one where the dependence structure of θ is not translation invariant. A shift on the log scale also fails.

Ultimately the challenge here is that the augmented posterior distribution contains additional information, which in some sense should not be counted when determining uncertainty in g^*. For the examples in the remainder of this paper we will just shift the point estimates.

1.5. Supplementing statistical graphics with sound

For the benefit of vision-impaired researchers, tools have recently been developed that engage sound, smell, touch, muscular resistance, voice dialogue, balance, and multiple senses at once (Dayé and de Campo 2006; Bornmann 2024; Sturdevant, Godfrey, and Gelman 2022; Gelman and Sturdevant

2023). Here we discuss a different use case in which sound is used to supplement a dynamic visual display.

Sound has a much lower information bandwidth than vision, but it has the advantage of not requiring attention. An audible signal can reach us even when our eyes and hands are occupied with dynamic graphics. In the present paper, we develop a visual dashboard and sonic feedback to facilitate the exploration of fitted models by allowing the user to perturb the model parameters, with the feedback conveying when and where the fit is getting worse. The goal is for the user to develop a better understanding of where the model fits the data, where it doesn't, and why a better fit is not possible.

2. Examples

2.1. A two-parameter model for golf putting

We begin with an example where a subtle model misfit can be identified by a loud squeal caused by forcing the curve to go through a pseudodata point that is just a little bit off of the best-fit curve.

We start with a two-parameter model from Broadie (2018) and Gelman (2019) presents the probability of a success of a putt in golf as a function of the distance x (measured in feet) from the hole:

$$g(x|\theta) = \left(2\Phi\left(\frac{\sin^{-1}((R-r)/x)}{\sigma_{\text{angle}}}\right) - 1\right)\left(\Phi\left(\frac{2}{(x+1)\,\sigma_{\text{distance}}}\right) - \Phi\left(\frac{-1}{(x+1)\,\sigma_{\text{distance}}}\right)\right), \quad (2)$$

where Φ is the normal cumulative distribution function; R and r are the radius of the hole and the ball; and $\theta = (\sigma_{\text{angle}}, \sigma_{\text{distance}})$ are the standard deviation of the variation in the angle and the log distance of the shot, relative to the aimed angle and distance.

We fit this model to a database of 1.25 million attempted putts from pro golfers, assuming the probability of success depends only on the distance from the hole. It would make sense to later fit a hierarchical model allowing this probability to vary by player, weather, and characteristics of the golf course, but it is reasonable to start with aggregate data to get a sense of the overall model fit.

Distances from the hole are binned into 31 categories i, centered at distances x_i of 0.25, 1, 2, 3, ..., with the bins becoming wider at greater distances and with the final bin centered at 75 feet. For each bin, we fit the model,

$$n_i y_i \sim \text{binomial}(n_i, g(x_i|\theta)), \quad (3)$$

where n_i is the number of shots from that distance bin and y_i is the proportion of those shots that went in.

Figure 4 shows the data $(x, y)_i,\ i = 1, \ldots, N$, along with the best-fit curve $g(x|\hat{\theta})$. With over a million data points and only two parameters, the model is estimated so precisely that curves corresponding to posterior simulation draws, $g(x|\theta^s)$ would be visually indistinguishable from the best fit.

It is impressive how well this two-parameter curve lines up with the data—but the fit is not perfect. There is a region between $x = 10$ and $x = 50$ where the curve is too low, and we would like to pull it upward. This is a job for the Squealer!

Because we have not yet set up the dynamic graphics, we perturb the curve by trial and error. First we grab the value of $g(x)$ at $x = 20$ and move it upward to match the data; the result is shown in the top row of Figure 5. This fixes the misfit between $x = 20$ and $x = 50$ but creates a new problem for x between 5 and 15. But we can fix this by tying the curve to two pseudo-data points, as demonstrated in the bottom row of Figure 5.

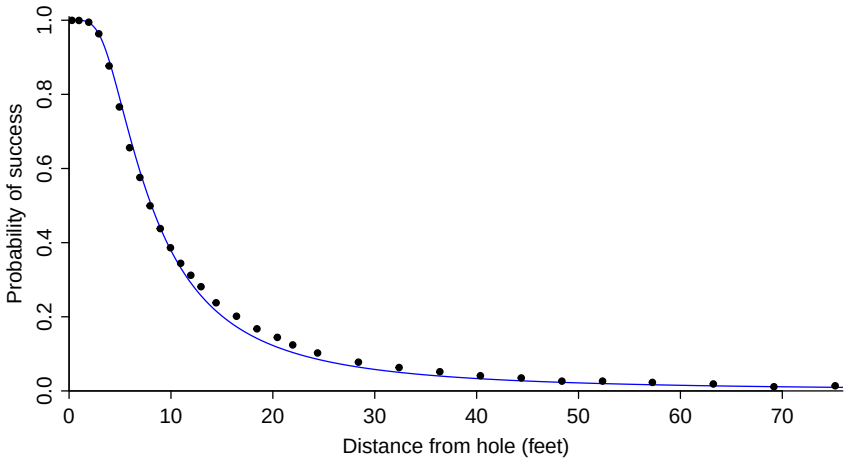


Figure 4: *Data on the proportion of successful golf putts as a function of distance from the hole, along with the curve corresponding to the best-fit estimate from a specified two-parameter model. The model fit shows some problems for distances between 10 and 50 feet, and we would like to explore what happens if we try to move the curve toward the data.*

We next turn to the Squealer dashboard and point-by-point summary to understand the statistical price paid by the apparently better fit. Figure 6 shows the result. The new curve $g(x|\theta^*)$ looks good but it pays a huge price in the likelihood, coming from a few points on the left of the graph. What is happening is that there are hundreds of thousands of putt attempts in these first few bins (consider that the short putts in the dataset include the failed attempts from other distances that came close but did not go in, hence there is a pileup of data near $x = 0$), hence the binomial likelihood is very strong there, so much that it does not easily tolerate even very small departures of the data y_i from their expected values $g(x_i|\theta)$.

No model is perfect, and it is not ideal for the fit of the entire curve to be driven so strongly by these few data points. Gelman, Vehtari, McElreath, et al. (2026) extend the model by adding an error term to account for imperfections in (2) and (3); our point here is to demonstrate how the Squealer can both facilitate a better fit to data and give insight into why the initial fitting procedure did not get there.

We include a software prototype with the golf example at `https://github.com/elc45/squealer-prototype/`. In this interface, the starting display shows the data superimposed with the curve of the posterior mode fit. The user can click the plot to add a pseudo-data point at the location of the cursor, optionally using the sliding scale at the bottom of the console to control the "gravitational pull" of the point, and the curve will automatically adjust to be the mode of the new posterior augmented with the pseudo-data point. The program then makes a noise whose pitch is proportional to the decrease in log posterior *under the original model* incurred by this change in fit. Under the hood, this new pseudo-mode is found in real time using the L-BFGS optimizer. The user can subsequently add more points in different locations, adjust the gravitational pull, or remove points, with the curve responding in real time.

2.2. A model for calibration in bioassays

We next show an example of the Squealer applied to a hierarchical data structure. Gelman, Chew, and Shnaidman (2004) present a Bayesian approach to inference from laboratory assays in which

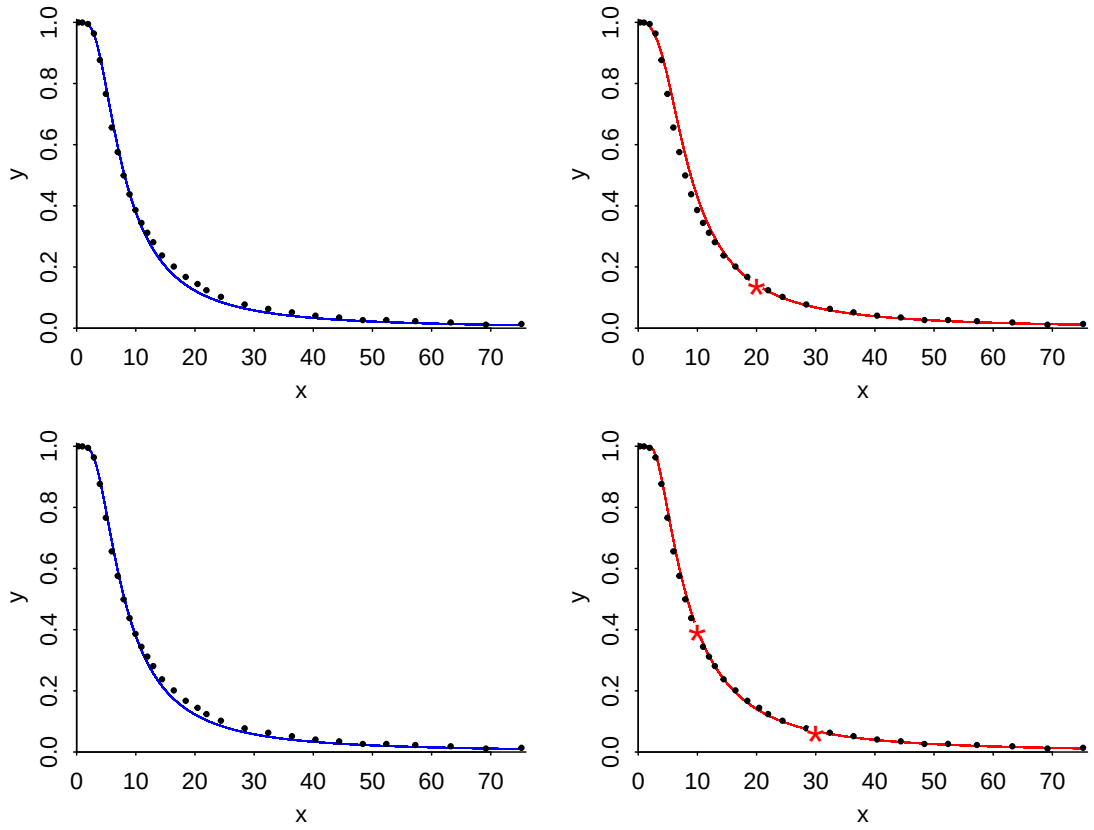


Figure 5: Top row: *first try to adjust the golf model to line up with the data, obtained by dragging the curve upward at one point at* $x = 20$. Bottom row: *second try, dragging the curve at two points* ($x = 10$ *and* $x = 30$)*, yields a much improved visual fit to the data. Point-by-point details are shown in Figure 6. These graphs are intended to display multiple curves corresponding to posterior simulation draws of* θ*, but in this case the posteriors are so precisely determined that no variation in the curves is visible.*

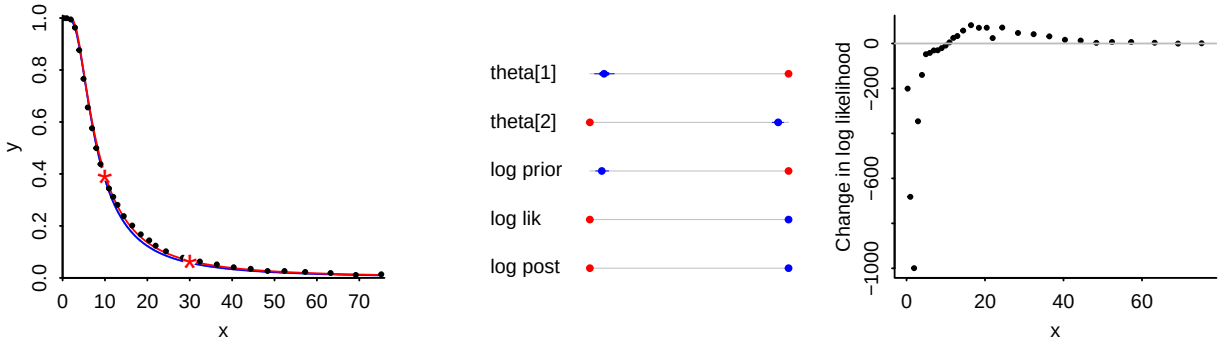


Figure 6: *Squealer applied to the golf model. It is possible to pull the fitted curve* $g(x|\hat{\theta})$ *toward two pseudo-data points and obtain a new curve* $g^* = g(x|\theta^*)$ *that is a better visual match to the data, but the log-posterior density for* θ^* *is much lower. The rightmost plot shows that the poor fit is coming from the data points with the lowest values of* x*.*

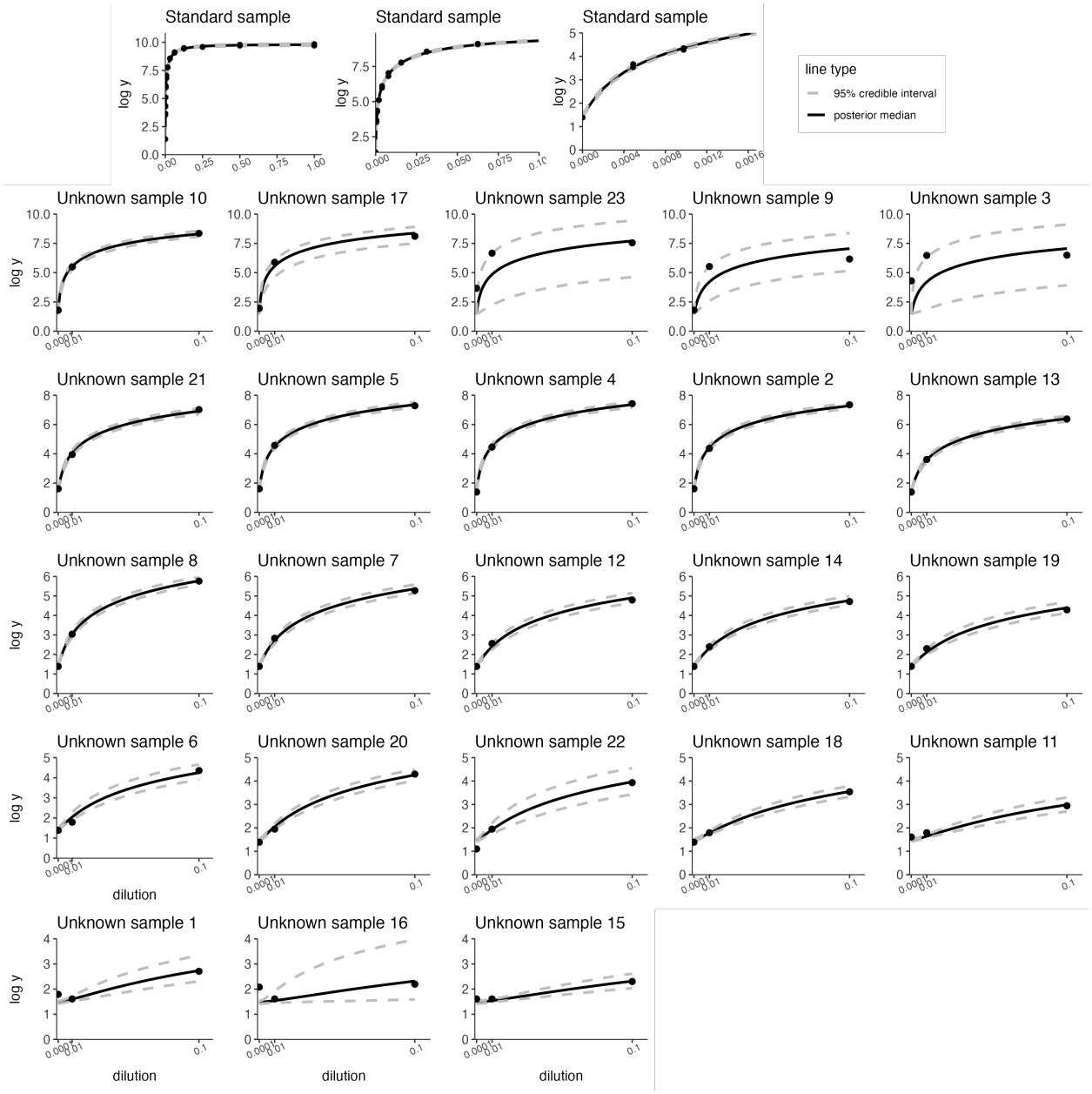


Figure 7: *A model for laboratory assays fit to calibration data (top row) and samples with unknown concentrations (remaining rows). For each, data have been gathered at multiple dilutions, and the curves show expected measurement value as a function of the dilution level. The unknown samples are displayed in decreasing order of mean measurements. The curves for samples 23 and 3 show some misfit to the data; we explore this with the Squealer in Figure 8.*

a model is simultaneously fit to calibration data (from samples with known concentrations) and unknown samples whose concentration is then estimated. More recently we have expanded the model by allowing for contamination. The challenge is that sometimes a single model does not appear to fit all the data well. Figure 7 shows an example of a plate with several dilutions of the calibration data and 22 unknown samples. The curves for samples 23 and 3 show some misfit to the data; we will explore this with the Squealer.

For each of samples 23 and 3 in Figure 7, we add three pseudo-data points (indicated by red asterisks on the plots for those samples) and re-fit the model. Figure 8 shows the result, with the original fit shown in black and the new fit in red. All the parameter estimates in the model change, but the differences in the fitted curves and their uncertainties are imperceptible except for the two affected samples.

It seems, then, that it is possible to get an improved (if imperfect) fit for those two unknown samples without interfering with the overall model fit, so where is the problem? Why did the model not fit this way originally?

The answer to this question can be seen in Figure 9, which displays the inferences for the parameters in the model that explain the differences between the samples. After applying the Squealer with the pseudo-data, the estimated concentrations for samples 23 and 3 are much bigger, and the set of 23 values of the parameter β_2 no longer fit a tight lognormal distribution. The penalty is being paid in the prior, with the cost being a higher group-level variance in the hierarchical model for the β_2's. Also the posterior uncertainties for those two values are much smaller; this is an unfortunate artifact of the pseudo-data implementation, as discussed in Section 1.4.

2.3. A model of power spectra in cosmology

In the standard model of cosmology, most of the information is contained in the two-point function (power spectra or correlation functions) of astrophysical observables such as the number density of galaxies in a set of observed voxels or the intensity of the cosmic microwave background (CMB) measured at pixels on the sphere of the sky (e.g., Hu, Sugiyama, and Silk 1997; Dodelson 2003; Planck Collaboration 2020b). These spectra or correlation functions are completely determined by the vector of underlying cosmological parameters θ (e.g., H_0, the Hubble constant which describes the present-day expansion rate; $\Omega_{\rm c}h^2$ and $\Omega_{\rm b}h^2$, composite quantities which describe the densities of the cold dark matter and baryons; $A_{\rm s}$ and n_s which describe the distribution of matter in the universe at early times; τ, which describes the effect of the earliest stars and galaxies on the ionization state of gas in the universe). For quantities measured on the sphere of the sky, such as the CMB, the power spectrum $C_\ell(\theta)$ determines the statistical properties of the observables, with a distribution that is typically well-described by a multivariate normal distribution. Here, ℓ denotes the spherical harmonic multipole, the equivalent of the Fourier mode k. The power spectrum is measured at spherical harmonic multipoles $2 \leq \ell \leq \ell_{\rm max}$ as $\hat{C}_\ell$. The quantity $C_\ell(\theta)$ depends on the cosmological parameters in a complicated, nonlinear way. Formulated in our framework, ℓ is equivalent to x, $g(x|\theta)$ to $C_\ell(\theta)$ and y to $\hat{C}_\ell$. Because of this dependence, this gives us a posterior distribution

$$p(\theta|d) \propto p(\theta)p(d|\theta) = p(\theta)p\left(d|C_\ell(\theta)\right). \tag{4}$$

As a slight subtlety, this posterior can be realized in several different ways. The most self-consistent (Bayesian) algorithm is to consider the multivariate normal likelihood corresponding to the model

$$d_p \sim \mathrm{MVN}(0, \Sigma_{pp'} + N_{pp'}) \tag{5}$$

where the data are measured at pixels $p = 1, \ldots, n_{\rm pix}$, and $\Sigma_{pp'}$ and $N_{pp'}$ are covariance matrices:

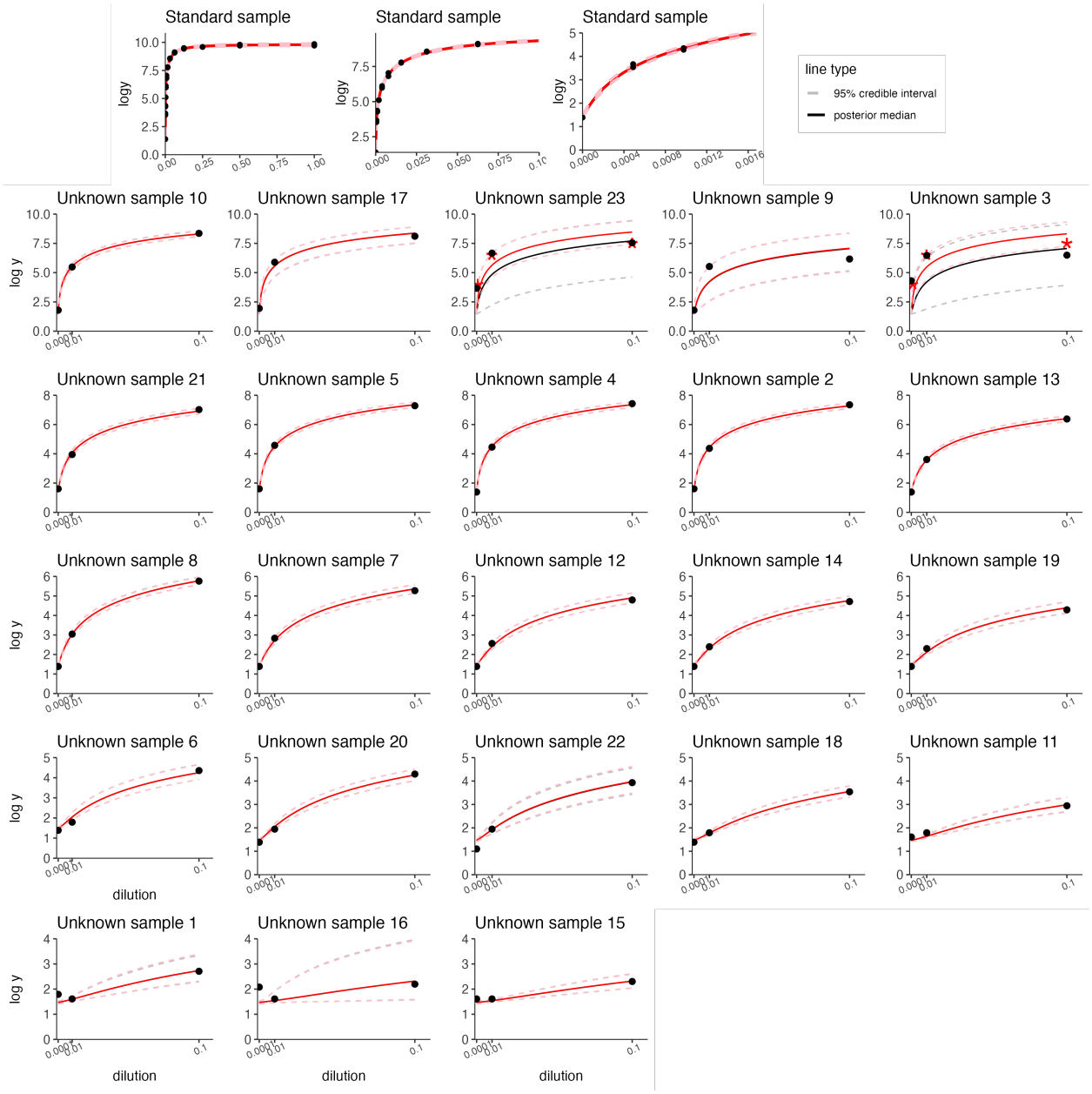


Figure 8: *Applying the Squealer to the data and model from Figure 7 to perturb the curves for samples 23 and 3. The curves fit to the original data are in black; the Squealer results are in red. Except for the two affected samples, the two fits are essentially identical. The next step is to understand what was constraining the original fit to be off for samples 23 and 3.*

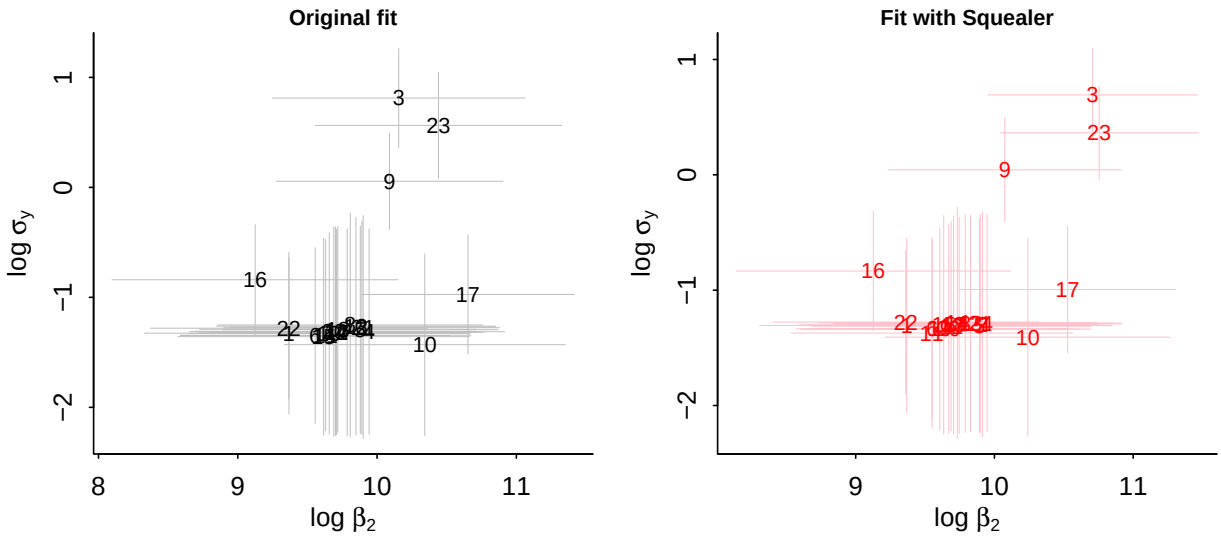


Figure 9: *Parameter estimates corresponding to the fitted models in Figures 7 and 8. The model has two parameters that vary across the 23 unknown samples: β_2, which is proportional to the concentration of the compound of interest within the sample, and σ_y, the scale of modeling/measurement error. The plots show the posterior estimates ± 1 standard error for $\log \beta_2$ and $\log \sigma_y$ for each sample. Perturbing the fit altered the inferences for samples 3 and 23 while barely changing anything for the other samples. Note the different scales of the two graphs.*

$\Sigma_{pp'}$ depends linearly on $C_\ell(\theta)$ and $N_{pp'}$ is a fixed noise covariance matrix, often well-approximated as diagonal. This is an accurate expression for our model of the underlying CMB field, but can be very expensive to compute, scaling as $O(n_{\text{pix}}^3)$.

In many cases, we consider the multivariate mode of $p(d|C_\ell)$ as estimates $\hat{C}_\ell = \hat{C}_\ell(d)$, and the likelihood can then be written as $p(\hat{C}_\ell|C_\ell)$. The shape around this mode can itself be approximated as a multivariate normal, although more precise approximations are sometimes available (Bond, Jaffe, and Knox 2000). Alternatively, the mode itself can be approximated at much lower computational cost as an unbiased estimator using the inverse spherical harmonic transform (Hivon et al. 2002; Alonso, Sanchez, and Slosar 2019).

These developments let us treat this as a curve-fitting problem to be visualized using the Squealer. Pseudo-data enforcing $C_{\ell^*} = \widetilde{C}_{\ell^*}$ (at some particular multipole $\ell = \ell^*$) can always be added as a new term in the likelihood $(C_{\ell^*} - \widetilde{C}_{\ell^*})^2/(2\bar{\sigma}^2_{\ell^*})$, no matter the form of the original likelihood.

Figure 10a shows measurements of the CMB power spectrum (Planck Collaboration 2020b). For $\ell < 30$, the spectrum is calculated as the mode of the full posterior (4), whereas for $\ell \geq 30$ it is calculated by inverse spherical harmonic transform and binned in ℓ for display. The red curve shows the best fit model, calculated as the posterior mode. For these calculations, we use the `plik-lite` likelihood for $\ell > 30$ and the `Commander` likelihood for $\ell \leq 30$, in both cases only using temperature data (these codes are described in Planck Collaboration 2020a). We model the effect of polarization data by including an informative prior on the parameter $\tau \sim \text{normal}(0.0543, 0.0073)$.

We can immediately see that there are regions where the residuals with respect to the curve, shown in the bottom panel, differ systematically from the data, for example the deficit of power around $\ell \simeq 20$ and the positive and negative excursions around $\ell \simeq 450$. What happens if we force the curve to go through these regions? In this case we care not only about the goodness of fit

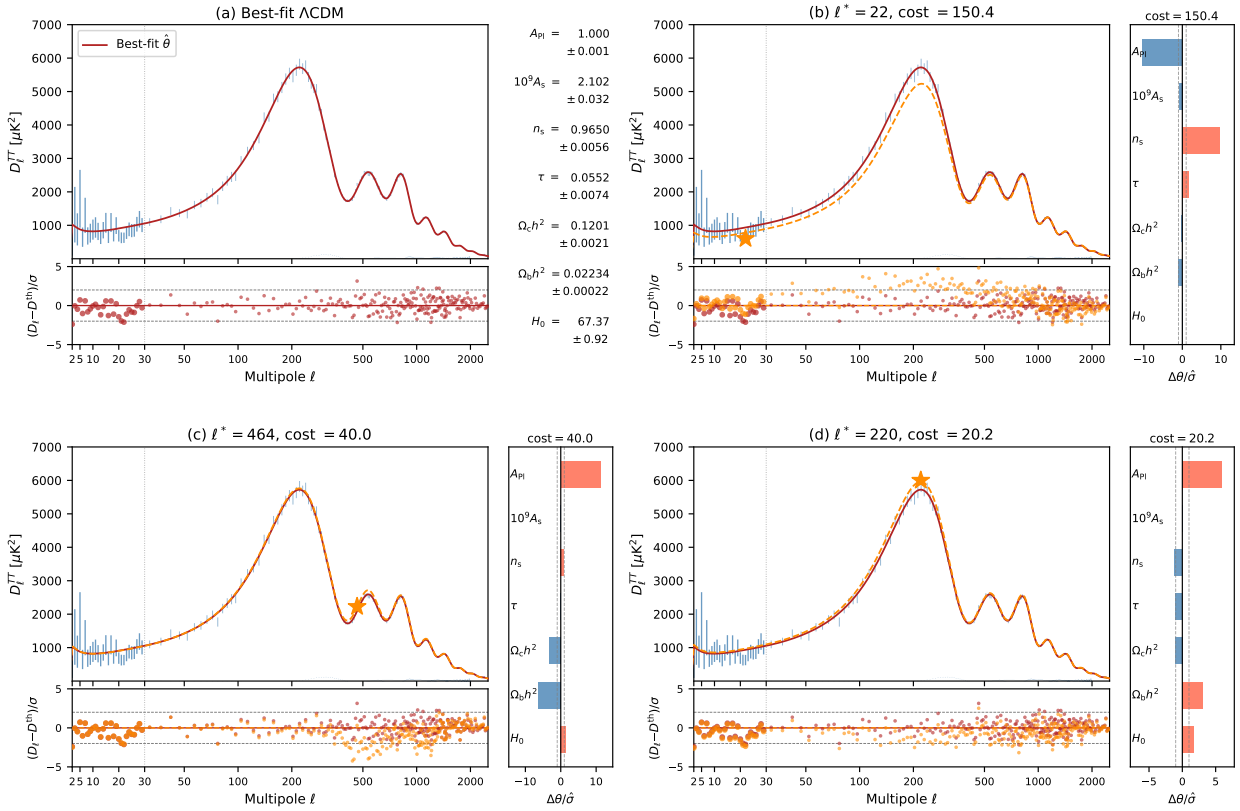


Figure 10: *The Squealer applied to the CMB power spectrum as measured by the Planck satellite. The x-axis is on an idiosyncratic scale to more evenly display the data. Here,* $D_\ell = \ell(\ell+1)C_\ell/(2\pi)$ *as is traditionally plotted in cosmology. Panel (a) shows the data (blue error bars), the overall best-fit (posterior mode) spectrum (red curve) and residuals between the two (red points in bottom panel, standard error units), along with inferred parameter values. Panels b, c, and d show an application of the Squealer. The orange star shows the pseudo-data point, the orange curve shows the new maximum posterior, with orange points in the bottom panel showing residuals. The right sub-panel shows the shift in each of the parameters (in standard error units), and is labeled by the cost, the decrease in the log posterior induced by the addition of the pseudo-data.*

when new pseudo-data are added, but the induced values θ^* are physically meaningful and might have relevance for other astrophysical observations. We show values of seven parameters being fit: A_{Pl} is an instrumental nuisance parameter describing the relative amplitude of different photon frequency measurements measured by the Planck satellite; the other six are parameters describing the cosmological model (Planck Collaboration 2020b).

The other three panels of Figure 10 show the Squealer applied to different regions of the plot. (Unlike the other applications discussed here, we show only the new best fit, rather than a set of samples.) Figure 10b shows an attempt to fit to the dip at $\ell \simeq 20$; there are no parameter values which modify the spectrum only in this region, and so the total cost to such a change is substantial. Figure 10c shows a modification at $\ell = 464$, which induces less drastic changes to the spectrum and to the parameters but requires larger negative excursions to accommodate the pseudo-data. As a further example, not actually favored by the data, Figure 10d shows how modifying the spectrum at exactly the first peak can be accommodated at even lower cost. In the latter two cases, known physical processes account for the ability to remain a relatively good fit.

2.4. A nonparametric Gaussian process model for temperature recordings

Gaussian processes (GPs) are commonly used for non-parametric modeling of time series data (Rasmussen and Williams 2005). In a typical setting, a researcher or practitioner is interested in recovering an unknown function f from noisy samples from that function. A GP distribution is used as a prior for the unknown function that enforces domain knowledge about the smoothness of the f. Specifically, a simple GP model uses a normal observation model with fixed and (possibly unknown) residual standard deviation and a GP prior on the unknown function.

Here, we use this model in a simulated example, to estimate an individual's body temperature from noisy thermometer recordings. This example has the benefit that it's a commonly-used, easily interpretable model, and it's also straightforward to implement a Squealer—the pseudo-data point method is available in closed form when hyperparameters are fixed.

The simulated data is thermometer recordings taken every few minutes for several hours from one (potentially sick) person. The readings are noisy (see Figure 11) and we use a Gaussian process model to estimate the true temperature.[1] Specifically, we use the following model:

$$y \sim \text{normal}(f(t), \sigma) \tag{6}$$

$$f(t) \sim \mathcal{GP}(98.6, k), \tag{7}$$

where k is the squared exponential kernel (Rasmussen and Williams 2005),

$$k(t, t') = \alpha \exp\left(\frac{-(t-t')^2}{2\ell^2}\right). \tag{8}$$

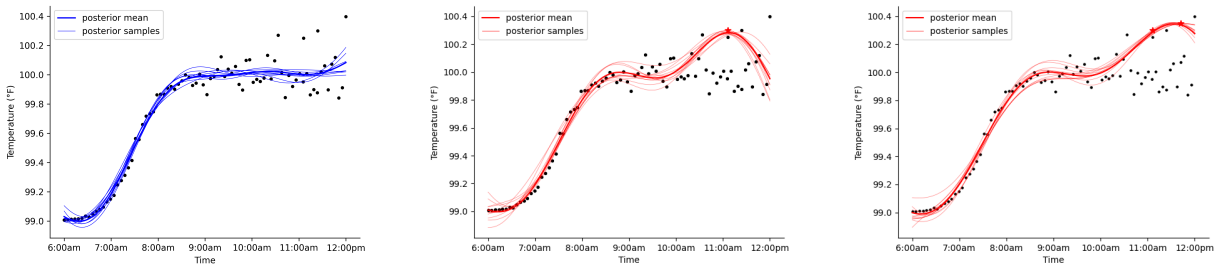
We fit the hyperparameters of the model—the variance α, the length scale ℓ, and the residual standard deviation σ—by maximizing the log marginal likelihood. That is, assuming $\theta = (\alpha, \ell, \sigma)$, we maximize the function (defined up to an additive constant)

$$\log p(y|\theta) = -\frac{1}{2} y^T (K(\theta) + \sigma^2 I)^{-1} y - \frac{1}{2} \log |K(\theta) + \sigma^2 I| \tag{9}$$

over θ.

After fitting hyperparameters, we sample from the GP, conditional on the observed data, fitted hyperparameters, and residual standard deviation; see Figure 11a. The posterior draws look

[1] We code the GP by hand in Python, using elementary functions from `torch`.



(a) original (b) one pseudo-data point (c) two pseudo-data points

Figure 11: *Fitted curves corresponding to draws from the GP posterior with pseudo-data, fixing the fitted hyperparameters of the covariance kernel.*

	ℓ	α	σ
original	0.23	0.52	0.09
one pseudo-point	0.23	0.48	0.15
two pseudo-points	0.26	0.58	0.17

Table 1: *Fitted values of hyperparameters, timescale ℓ, variance α, and residual standard deviation σ for the original model and for both versions with pseudo-data.*

reasonable, other than the large observation noise near 6am and the small observation noise near noon. The posterior draws estimate a temperature around 100°F after around 9am. However, a possible concern is the upward-sloping temperature recordings above 100.2° F towards the end of the interval. The posterior doesn't place much mass near these points, but a nervous doctor might want to get a sense why. Does the Gaussian process prior not allow it? Are there too few recordings there? How "easy" would it be to move the posterior through these points? To get a sense of the answer for these questions, we use the Squealer.

Adding the pseudo-data point (x^*, y^*), we now maximize a new log marginal likelihood (now marginalizing over $N+1$ instead of N dimensions) where

$$y \to \begin{bmatrix} y \\ y^* \end{bmatrix}, \qquad K(\theta) \to \begin{bmatrix} K(\theta) & k(x^*, x) \\ k(x^*, x)^T & k(x^*, x^*) \end{bmatrix}, \qquad \sigma^2 I \to \begin{bmatrix} \sigma^2 I & 0 \\ 0 & 10^{-6} \end{bmatrix}. \tag{10}$$

Here, $k(x^*, x)$ is a column vector whose i^{th} element is $k(x^*, x_i)$. We use a variance of 10^{-6} for the pseudo-data point which has the effect of strongly dragging the posterior through (x^*, y^*).

In Figure 11b, we add a pseudo-data point of 100.3°F at 11am, adjusting the posterior so that it moves through the pseudo-data point in a smooth way. But the posterior still doesn't "want to" go through the points over 100.2°F. The smoothness imposed by the squared-exponential kernel still allows the posterior to return to its former posterior mean (without a pseudo-data point) by noon, the end of the interval. When we add a second pseudo-data point (see Figure 11c), the posterior now goes roughly through the points above 100.2°F.

Table 1 shows the fitted hyperparameter values for the original posterior, and the posteriors with one and then two pseudo-data points. The residual standard deviation, σ, increases from 0.10 to 0.14 to 0.17 when going from zero to one to two pseudo-data points. So the model is allowing the posterior to go through the pseudo-data points by fitting a larger observation noise. In other words, in order to get the posterior to go through the high temperature points, the pseudo-data points forced the posterior observation noise to be larger than the original data indicated. The

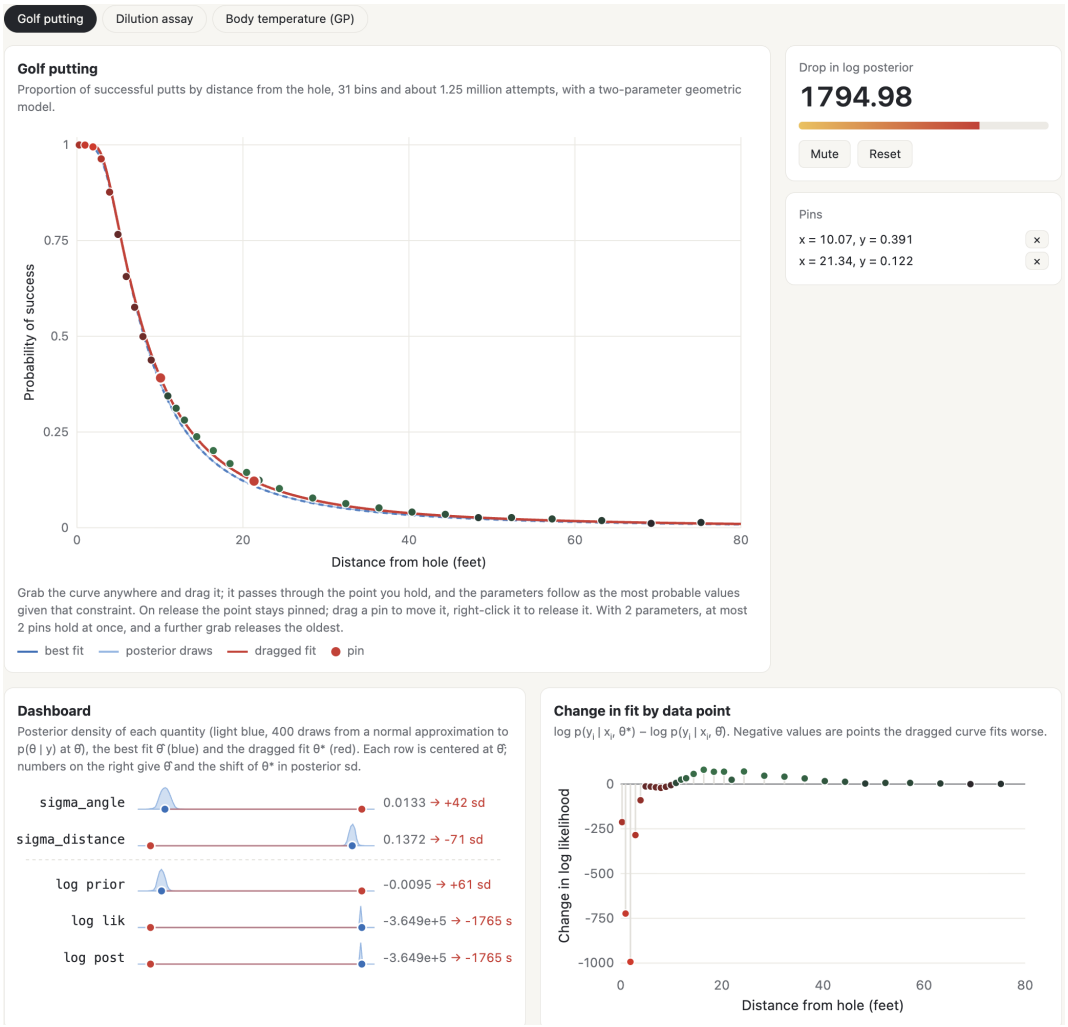


Figure 12: *Golf example in our Squealer web application showing the display of data and fitted curves, parameter dashboard, misfit in aggregate and for each datapoint, and a record of the pseudo-data points added to the fit.*

combination of the normal observation model and a large number of measurements just less than 100°F makes the original posterior concentrate on a smaller residual standard deviation and puts little weight on high temperatures.

3. Implementation

We developed the principles of the Squealer with static graphics, but ultimately it is designed to operate dynamically, so that a user can grab, move, and pin a fitted curve, see the consequences for the estimated parameters and the fit to individual data points, and hear the decline in the overall fit. Several challenges remain to reach that goal.

First, with a tool that's meant to be interactive, the program needs to quickly converge to the updated best fit or posterior distribution quickly to allow smooth operation. Using existing software this is done by nested loops, with the outer loop being the steps that move the curve as it is being dragged, and the inner loop being the iterations of Stan (or some other probabilistic

program) fitting the model at each step along that path. It would make more sense to thread these loops, so that the target distribution is gradually shifting while the simulations are happening, in the manner of sequential Monte Carlo (Liu and Chen 1998).

Second, details of the Squealer will depend on the problem under study. In the same way that maximum likelihood, variational inference, or Bayesian inference are general methods that have different implementations for different models, so must the computation and visual display of the Squealer be adapted to the mathematical structure and goals of the models being fit. We suspect that the best way to develop the Squealer will be to begin by setting it up for particular classes of models and then ultimately designing a more general implementation.

Third, we would like the different parts of the dashboard to work in concert, so that, in addition to being able to grab and pin fitted curves and see how these shift the fitted curves and change the fit to individual data points, the user can also shift and pin parameter values directly. Again, it will make the most sense to first implement this for a particular example.

Fourth, as discussed in Section 1.4, the gravitational implementation of the Squealer causes the posterior distribution to be constricted in the range of the pseudo-data points. We would like to develop a method for shifting the posterior that does not suffer from this artifact.

As a start, we developed a demo Squealer web tool and implemented it for the examples in Sections 2.1, 2.2, and 2.4 of this paper (excluding the cosmology example in Section 2.3 because it was not so conveniently encapsulated). Figure 12 shows a screenshot. The application runs entirely in the browser and starts with data and a fitted curve. The user then can grab the curve at any location and drag to move it. As the curve moves, the parameters of the model are recomputed in real time as the posterior mode subject to the curve passing exactly through the held point (details below). The penalty, or the drop in log posterior density, is reflected in the audio (squeal) which plays as the curve is moved. The higher the pitch, the larger the penalty. On mouse release the constrained point is pinned and used as a constraint in further user moves. Unlike the soft pull toward pseudo-data points described in Section 1.2, the constraint is exact, so the curve goes where the user moves it.

To find the new parameter values at each step of the movement, a constrained optimization problem is solved where we maximize the log posterior density subject to the curve passing exactly through the held point and other pinned points. For this we use an augmented Lagrangian method that is a soft pull (Section 1.2) plus a correction term. At each iteration, the correction term is adjusted by the amount the curve still misses the points, and the fit is repeated until the miss is negligible. Each step of the drag starts with the solution of the previous step so only a few further iterations are needed. For the Gaussian process example, no iteration is needed, since conditioning on a pinned point is exact and in closed form. Ultimately we would like a Bayesian simulation-based approach that expresses posterior uncertainty but this will require computation that is fast enough to allow the inference curve to be moved by the user in real time.

4. Discussion

The Squealer uses sonic feedback to supplement rather than replace a visual display. Often it is not immediately clear why a model is not fitting a dataset as the practitioner expects, or why the model is neglecting some fit that seems obvious to the user's eye. Even with very expressive models like Gaussian processes, the factors constraining how they can fit data are opaque, especially for the non-expert. Allowing the modeler to make the curve go where they think it should can sometimes resolve this confusion, for instance by demonstrating that certain other points are very poorly fit by the new curve. But the likelihood may suffer in ways that are not visually apparent. Supplementing

the visual fit with auditory feedback can give the modeler helpful intuition as to why the model behaves as it does.

The sensification could also be done in different ways. One appealing idea is haptic feedback, where the user would use a joystick to drag the curve, and moving it away from its best fit would induce resistance on the controller so that the user would physically feel the resistance to degrading the fit to prior and data.

Our larger goal is to be able to look under the hood of models better to see why they don't fit as we'd expect. Specifically, we want to see which parameters suffer from the fit we want to impose on the model, and why. This could be a valuable step in the statistical and machine-learning workflow in the modern computing environment in which probabilistic programming tools allow us to fit increasingly complicated models.

Acknowledgments

The first two authors did this work as visiting scholars at the Flatiron Institute. We also thank Sam Felgran, Riley Carlin, Nidhi Ram, Siquan Wang, and an anonymous reviewer for helpful collaboration and the U.S. Office of Naval Research for partial support of this work.

Data and code availability

All the data and code for this article are freely available and archived with the article. Code for the demo is available at `https://github.com/magland/squealer-demo/`.

References

Alonso, David, Javier Sanchez, and Anže Slosar (2019). "A unified pseudo-C_ℓ framework". In: *Monthly Notices of the Royal Astronomical Society* 484, pp. 4127–4151.

Bond, J. R., A. H. Jaffe, and L. Knox (2000). "Radical compression of cosmic microwave background data". In: *Astrophysical Journal* 533, p. 19.

Bornmann, Lutz (2024). "The sound of science". In: *EMBO Reports* 25, pp. 3743–3747.

Broadie, Mark (2018). "Two simple putting models in golf". In: `https://statmodeling.stat.columbia.edu/wp-content/uploads/2019/03/putt_models_20181017.pdf`.

Dayé, Christian and Alberto de Campo (2006). "Sounds sequential: sonification in the social sciences". In: *Interdisciplinary Science Reviews* 31, pp. 349–364.

Dodelson, Scott (2003). *Modern Cosmology*. San Diego: Academic Press.

Gelman, Andrew (2004). "Exploratory data analysis for complex models (with discussion)". In: *Journal of Computational and Graphical Statistics* 13, pp. 755–787.

Gelman, Andrew (2019). "Model building and expansion for golf putting". In: *Stan Case Studies* 6. `https://mc-stan.org/users/documentation/case-studies/golf.html`.

Gelman, Andrew (2020). "The typical set and its relevance to Bayesian computation". In: *Statistical Modeling, Causal Inference, and Social Science*. 2 Aug. `https://statmodeling.stat.columbia.edu/2020/08/02/the-typical-set-and-its-relevance-to-bayesian-computation/`.

Gelman, Andrew, Ginger Chew, and Michael Shnaidman (2004). "Bayesian analysis of serial dilution assays". In: *Biometrics* 60, pp. 407–417.

Gelman, Andrew and S. Gwynn Sturdevant (2023). "From visualization to sensification". In: *Amstat News* 547, pp. 18–19.

Gelman, Andrew, Aki Vehtari, Richard McElreath, et al. (2026). *Bayesian Workflow*. London: CRC Press.

Hivon, E., K. M. Górski, C. B. Netterfield, B. P. Crill, S. Prunet, and F. Hansen (2002). "MASTER of the cosmic microwave background anisotropy power spectrum: A fast method for statistical analysis of large and complex cosmic microwave background data sets". In: *Astrophysical Journal* 567, p. 2.

Hu, Wayne, Naoshi Sugiyama, and Joseph Silk (1997). "The physics of microwave background anisotropies". In: *Nature* 386, pp. 37–43. eprint: `astro-ph/9504057`.

Liu, Jun S. and Rong Chen (1998). "Sequential Monte Carlo methods for dynamic systems". In: *Journal of the American Statistical Association* 93, pp. 1032–1044.

Planck Collaboration (2020a). "Planck 2018 results. V. CMB power spectra and likelihoods". In: *Astronomy & Astrophysics* 641, A5.

Planck Collaboration (2020b). "Planck 2018 results. VI. Cosmological parameters". In: *Astronomy & Astrophysics* 641, A6.

Rasmussen, Carl Edward and Christopher K. I. Williams (2005). *Gaussian Processes for Machine Learning*. MIT Press.

Sturdevant, S. Gwynn, A. Jonathan R. Godfrey, and Andrew Gelman (2022). "Delivering data differently". In: `https://arxiv.org/abs/2204.10854`.

Wickham, Hadley (2006). "Exploratory model analysis with R and GGobi". In: `https://had.co.nz/model-vis/2007-jsm.pdf`.

Wickham, Hadley, Dianne Cook, and Heike Hofmann (2015). "Visualizing statistical models: Removing the blindfold". In: *Statistical Analysis and Data Mining* 8, pp. 203–225.