Quiz 2, STATS/DATASCI 531/631 W25

In class on 4/16, 2:30pm to 3:00pm

This document produces different random quizzes each time the source code generating it is run. The actual quiz will be a realization generated by this random process, or something similar.

This version lists all the questions currently in the quiz generator. The actual quiz will have one question sampled from each of the 6 question categories.

Instructions. You have a time allowance of 30 minutes. The quiz may be ended early if everyone is done. The quiz is closed book, and you are not allowed access to any notes. Any electronic devices in your possession must be turned off and remain in a bag on the floor.

For each question, circle one letter answer and provide supporting reasoning. If you run out of space, you may continue on the back of the page, but please indicate to the reader that you are doing so.

Q1. Foundations of POMP models

Q8-01.

Consider a model $Y_{1:N}$ for data $y^*_{1:N}$, with a latent variable $X_{0:N}$, and a statistical model defined by a joint density $f_{X_{0:N},Y_{1:N}}(x_{0:N},y_{1:N};\theta)$. The likelihood function is

$$L(\theta) = f_{Y_{1:N}}(y^*_{1:N};\theta).$$

Are the following identities (A) true for all statistical models; (B) true for general POMP models but not all models; (C) true for linear Gaussian POMP models but not general POMP models; (D) generally false.

$$L(\theta) = \int f_{Y_{1:N}|X_{0:N}}(y^*_{1:N}|x_{0:N};\theta)\, f_{X_{0:N}}(x_{0:N};\theta)\, dx_{0:N} \tag{1}$$

$$L(\theta) = \prod_{n=1}^{N} f_{Y_n|Y_{1:n-1}}(y^*_n|y^*_{1:n-1};\theta) \tag{2}$$

$$\mathrm{Var}\{X_{n+1}\,|\,Y_{1:n}\} = E[\mathrm{Var}\{X_{n+1}\,|\,X_n\}\,\big|\,Y_{1:n}] + \mathrm{Var}\{E[X_{n+1}\,|\,X_n]\,\big|\,Y_{1:n}\} \tag{3}$$

$$L(\theta) = \int \left[\prod_{n=1}^{N} f_{Y_n|X_n}(y^*_n|x_n;\theta)\right] f_{X_{0:N}}(x_{0:N};\theta)\, dx_{0:N} \tag{4}$$

Solution. 1(A), 2(A), 3(B), 4(B)
For part 3, we have in general that

$$\mathrm{Var}\{U\} = E[\mathrm{Var}\{U|V\}] + \mathrm{Var}E[U|V]. \tag{5}$$

Such identities hold when everything is conditioned on another variable, so we have

$$\mathrm{Var}\{U|Z\} = E[\mathrm{Var}\{U|V,Z\}|Z] + \mathrm{Var}\{E[U|V,Z]|Z\}. \tag{6}$$

Setting $U = X_{n+1}$, $V = X_n$, $Z = Y_{1:n}$, we have

$$\mathrm{Var}\{X_{n+1}\,|\,X_n, Y_{1:n}\} = E[\mathrm{Var}\{X_{n+1}\,|\,X_n, Y_{1:n}\}\,\big|\,X_n] + \mathrm{Var}\{E[X_{n+1}|X_n, Y_{1:n}]\,\big|X_n\}. \tag{7}$$

The POMP model structure assumes X_{n+1} and $Y_{1:n}$ are conditionally independent given X_n and so

$$\mathrm{Var}\{X_{n+1}|_n\} = E[\mathrm{Var}\{X_{n+1}|X_n\}|X_{1:n}] + \mathrm{Var}\{E[X_{n+1}|X_n]|X_{1:n}\}. \tag{8}$$

Q8-02.

Which of the following linear Gaussian POMP model have an observable variable Y_n with distribution matching an ARMA(1,1) model? Here, ϵ_n and η_n are Gaussian white noise. X_n is 1-dimensional in (1) and 2-dimensional in (2) and (3).

A. Only (3)
B. (1) and (2) but not (3)
C. (2) and (3) but not (1)
D. (1) and (3) but not (2)
E. (1), (2) and (3)

$$\left.\begin{array}{rcl} X_n &=& aX_{n-1} + \epsilon_n \\ Y_n &=& X_n + \eta_n \end{array}\right\} \tag{9}$$

$$\left.\begin{array}{rcl} X_n &=& \begin{pmatrix} a & 1 \\ 0 & 0 \end{pmatrix} X_{n-1} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \epsilon_n \\ Y_n &=& (1,0)\, X_n + \eta_n \end{array}\right\} \tag{10}$$

$$\left.\begin{array}{rcl} X_n &=& \begin{pmatrix} a & 1 \\ 0 & 0 \end{pmatrix} X_{n-1} + \begin{pmatrix} 1 \\ b \end{pmatrix} \epsilon_n \\ Y_n &=& (1,0)\, X_n \end{array}\right\} \tag{11}$$

Solution. A
A nice way to see Eq. (1) is in the frequency domain. X_n is AR(1) and so has spectrum $\sigma_\epsilon^2/|1-a\exp i\omega|^2 = \sigma_\epsilon^2/(1+a^2-2a\cos(\omega))$. η_n has spectrum σ_η^2, and so Y_n has spectrum

$$\begin{aligned} \sigma_\eta^2 + \frac{\sigma_\epsilon^2}{1+a^2-2a\cos(\omega)} &= \frac{\sigma_\eta^2(1+a^2)+\sigma_\epsilon^2-2a\sigma_e^2 ta\cos(\omega)}{1+a^2-2a\cos(\omega)} &(12)\\ &= \sigma^2\frac{(1+b^2)-2b\cos(\omega)}{1+a^2-2a\cos(\omega)} &(13)\end{aligned}$$

where σ and b solve

$$\begin{aligned} \sigma^2(1+b^2) &= \sigma_\eta^2(1+a^2)+\sigma_\epsilon^2, &(14)\\ \sigma^2 b &= a\sigma_\eta^2 &(15)\end{aligned}$$

Eq. (2) is the same set of equations as (1), as seen by setting $X_n' = \begin{pmatrix} X_n \\ \epsilon_n \end{pmatrix}$.

Eq. (3) is the ARMA(1,1) case of the LG-POMP representation of a general ARMA model givein in Chapter 11. Specifically, set $X_n = \begin{pmatrix} Y_n \\ b\epsilon_n \end{pmatrix}$, and see that the process model becomes

$$\begin{pmatrix} Y_n \\ b\epsilon_n \end{pmatrix} = \begin{pmatrix} aY_{n-1} + b\epsilon_{n-1} + \epsilon_n \\ b\epsilon_n \end{pmatrix}.$$

Q8-03.

Scientifically, our conclusions should not depend on the units of measurement we use, but we can make errors if we don't get the details right. Suppose our data are two years of weekly aggregated case reports of a disease and we have a continuous-time model solved numerically by an Euler timestep of size dt. Which of the following is a correct explanation of our options for properly implementing this in a pomp object called `po`?

A. The measurement times, `time(po)`, should be in units of weeks, such as $1, 2, \ldots, 104$. The latent process can be modeled using arbitrary time units, say days or weeks or years. The units of dt should match the time units of the **latent** process.
B. The measurement times, `time(po)`, should be in units of weeks, such as $1, 2, \ldots, 104$. The latent process can be modeled using arbitrary time units, say days or weeks or years. The units of dt should be in weeks (in practice, usually a fraction of a week) to match the units of the **measurement** times.
C. The measurement times do not have to be in units of weeks. For example, we could use `time(po)`=1/52, 2/52, ..., 2. The latent process and dt should use the same units of time as the measurement times.
D. The measurement times do not have to be in units of weeks. For example, we could use `time(po)`=1/52, 2/52, ..., 2. The latent process can also use arbitrary units of time, which do not necessarily match the units of the measurement times. The units of dt should match the units used for the **latent** process.
E. The measurement times do not have to be in units of weeks. For example, we could use `time(po)`=1/52, 2/52, ..., 2. The latent process can also use arbitrary units of time, which do not necessarily match the units of the measurement times. The units of dt should match the units used for the **measurement** times.

Solution. C
For scientific calculations, you generally have to pick an arbitrary set of units and use it consistently. In `pomp`, this means that you have to use the same units for measurement times and within the latent process. For example, if your measurement times are in days (7,14,...) then rate parameters should have units day^{-1}. A latent transition with mean duration 1 week would have corresponding rate $1/7\text{day}^{-1}$.

When you report your answers, you can use any appropriate units depending on the scale of the quantity—for example, mean life expectancy in years, mean infection duration in days, accumulated cases over a week or a month. Within the computer code, you are advised not to mix units. That is because computers are not good at representing units; a computer is good at representing a number, say 2.5, but usually our code does not know if this number corresponds to 2.5 days or weeks or years. 2.5 hours is the same as 150 minutes, but we can't just replace 2.5 by 150 in our code and get the same answer. That is in contrast to scientific work, where every numerical quantity should have its unit attached, and answers of 2.5hr and 150min are equally correct.

Q8-04.

Let V_n be a Markov process and let $W_n = h(V_n)$ for some function h. Let (X_n, Y_n) be a POMP with latent process X_n and observed process Y_n. Which of the following statements are true?

i. W_n is a Markov process for all choices of h.

ii. W_n is a Markov process for some choices of h.

iii. W_n is not a Markov process for any choice of h.

iv. If $V_n = (X_n, Y_n)$ and $h(X_n, Y_n) = X_n$ then W_n is a Markov process.

v. If $V_n = (X_n, Y_n)$ and $h(X_n, Y_n) = Y_n$ then W_n is a Markov process.

A. i,iv,v
B. ii,iv
C. ii,v
D. iii
E. Some combination other than those listed above

Solution. B
(iv) is true by definition of a POMP model, and this rules out (iii). To see that (v) is not generally true, consider a Markov chain, X_n on {1,2,3} which cycles $1 \to 2 \to 3 \to 1$, with X_0 being uniform on {1,2,3}. Let $Y_n = A$ if $X_n \in \{1, 2\}$ and $Y_n = B$ if $X_n = 3$. Then, $P(Y_{n+1} = B|Y_n = A) = 1/2$ which is not equal to $P(Y_{n+1} = B|Y_n = A, Y_{n-1} = A) = 1$. This example also rules out (i). (ii) is clearly satisfied by the identity function, $h(x) = x$, so we are left with (ii) and (iv).

Q2. Likelihood evaluation; the particle filter

Q9-01.

Suppose that 10 replications of a particle filter, each using 10^3 particles, runs in 15 minutes with no parallelization. To look for a more precise likelihood evaulation, you consider running 20 replicates, each with 10^4 particles. Approximately how many minutes will this take, if you distribute the calculation across 4 cores?

A. 50
B. 60
C. 75
D. 120
E. 300

Solution. C.
Using the linear dependence, also called proportionality, of the computing effort on various algorithmic parameters, we calculate

```
15 * (10000/1000) * (20/10) * (1/4)
```

```
## [1] 75
```

Q9-02.

A particle filter is repeated 5 times to evaluate the likelihood at a proposed maximum likelihood estimate, each time with 10^4 particles. Suppose the log likelihood estimates are -2446.0, -2444.0, -2443.0, -2442.0, -2440.0. Which of the following is an appropriate estimate for the log likelihood at this parameter value and its standard error.

A. Estimate = -2443.0, with standard error 1.0
B. Estimate = -2443.0, with standard error 2.2
C. Estimate = -2443.0, with standard error 5.0
D. Estimate = -2441.4, with standard error 2.2
E. Estimate = -2441.4, with standard error 1.4

Solution. E.
Answers A, B and C estimate using a mean on the log scale. However, the particle filter provides an unbiased likelihood estimate on a natural scale but not on a log scale. Note that the particle filter also has some bias for most quantities on a natural scale, which reduces to zero as the number of particles tends to infinity, but it happens to be unbiased for the likelihood. The standard error for the log of the mean of the likelihoods can be computed by the delta method or a jack-knife, for example using the logmeanexp function in pomp.

```
  ll <- c(-2446,-2444,-2443,-2442,-2440)
  mean(ll)
```

```
## [1] -2443
```

```
  sd(ll)
```

```
## [1] 2.236068
```

```
  sd(ll)/sqrt(length(ll))
```

```
## [1] 1
```

```
pomp::logmeanexp(ll,se=TRUE)
```

```
##          est           se
## -2441.422198     1.380688
```

Q9-03.

Effective sample size (ESS) is one of the main tools for diagnosing the success of a particle filter. If you plot an object of class `pfilterd_pomp` (created by applying `pfilter` to a pomp object), the ESS is displayed. Suppose one or more time points have low ESS (say, less than 10) even when using a fairly large number of particles (say, 10^4). What is the proper interpretation?

A. There is a problem with data, perhaps an error recording an observation.
B. There is a problem with the model which means that it cannot explain something in the data.
C. The model and data have no major problems, but the model happens to be problematic for the particle filter algorithm.
D. At least one of A, B, and C.
E. Either A or B or both, but not C. If the model fits the data well, the particle filter is guaranteed to work well.

Solution. D.
An example of a situation where the model fits the data well, but filtering is hard, arises when the measurement error is small relative to the process noise. In this case, the particles are scattered by the process noise and very few of them are compatible with the data due to the precise measurement. Thus, almost all the particles must be discarded as unfeasible given the data, corresponding to a low ESS.

Q9-04.

In a particle filter, a particle that is resampled k times is said to be the parent of these k children, The complete ancestry graph of all the particles is an evolutionary tree for the population of particles. Each filtering iteration corresponds to a generation of this population process. Darwinian evolution on populations occurs when individuals with the fittest phenotype reproduce more offspring for the next generation, and these children inherit the parent's genotype subject to mutation.

Particle filter	Darwinian evolution
(1) Prediction step: simulation	(a) Genotype
(2) Filtering step: resampling	(b) Fitness
(3) State value, $X_{n,j}$	(c) Mutation
(4) Measurement density, $f_{Y_n\|X_n}(y_n^*\|X_{n,j})$	(d) Reproduction

What is the pairing between the particle filter concepts (numbers 1–4) and the analogous evolutionary concepts (letters a–d).

A. (1a) (2b) (3c) (4d)
B. (1b) (2a) (3c) (4d)
C. (1c) (2d) (3a) (4b)
D. (1d) (2c) (3a) (4b)
E. A combination not listed above.

Solution. C.
Inherited similarities between generations is based on the genotype, which in a POMP model is the Markovian state. The mutation process that randomly generates the genotype of the child is therefore `rprocess`, the state simulation process. The expected number of children depends on the fitness, which here is the measurement density. The actual number of children is a random process, corresponding to the resampling of particles.

Q3. Likelihood maximization; iterated filtering

Q10-01.

When carrying out inference by iterated particle filtering, the likelihood increases for the first 10 iterations or so, and then steadily decreases. Testing the inference procedure on simulated data, this does not happen and the likelihood increases steadily toward convergence. Which of the following is the best explanation for this?

A. One or more random walk standard deviations is too large.
B. One or more random walk standard deviations is too small.
C. The model is misspecified, so it does not fit the data adequately.
D. A combination of the parameters is weakly identified, leading to a ridge in the likelihood surface.
E. Too few particles are being used.

Solution. C.
A test on simulated data, when the truth is known, can help pin down an optimization problem. All the issues other than C can cause inference problems, but likely would cause similar problems on simulated data.

When there is a reproducible and stable phenomenon of decreasing likelihood, it generally indicates that the unperturbed model is a worse fit to the data than the perturbed model. Recall that the likelihood calculated by iterated filtering at each iteration corresponds to the model with perturbed parameters rather than the actual postulated model with fixed parameters. If the perturbed model has higher likelihood, it may mean that the data are asking to have time-varying parameters. It may also be a signature of any other weakness in the model that can be somewhat accommodated by perturbing the parameters.

Q10-02.

People sometimes confuse likelihood profiles with likelihood slices. When you read a report claiming to have computed a profile it can be worth checking whether it is actually computed as a slice. Suppose you read a figure which claims to construct a profile confidence interval for a parameter ρ in a POMP model with four unknown parameters. Which of the following confirms that the plot is, or is not, a properly constructed profile confidence interval.

A. The CI is constructed by obtaining the interval of rho values whose log likelihood is within 1.92 of the maximum on a smoothed curve of likelihood values plotted against ρ.
B. The code (made available to you by the authors as an Rmarkdown file) involves evaluation of the likelihood but not maximization.
C. The points along the ρ axis are not equally spaced.
D. The smoothed line shown in the plot is close to quadratic.
E. A and D together.

Solution. B.
If the researchers calculate a sliced likelihood through the MLE and tell you it is a profile, but you are concerned they might have constructed a slice by mistake, it is hard to know without looking at the code. (A) is the proper construction of a profile if the points are maximizations over the remaining parameters for a range of fixed values of rho. However, if the code does not involve maximization over other parameters at each value of ρ, it cannot be a proper profile. It could be a slice accidentally explained to be a profile, and with a confidence interval constructed as if it were a profile.

If there is only one unknown parameter then a slice and a profile are the same thing, and no maximization is required. This is an unusual situation; there is usually more than one unknown parameter.

Q10-03.

The iterated filtering convergence diagnostics in figure 1 come from a student project investigating the market value of Gamestop. What is the best interpretation?

A. Everything seems to be working fine. The likelihood is climbing. The replicated searches are giving consistent runs. The spread of convergence points for σ_ν and H_0 indicates weak identifability, which is a

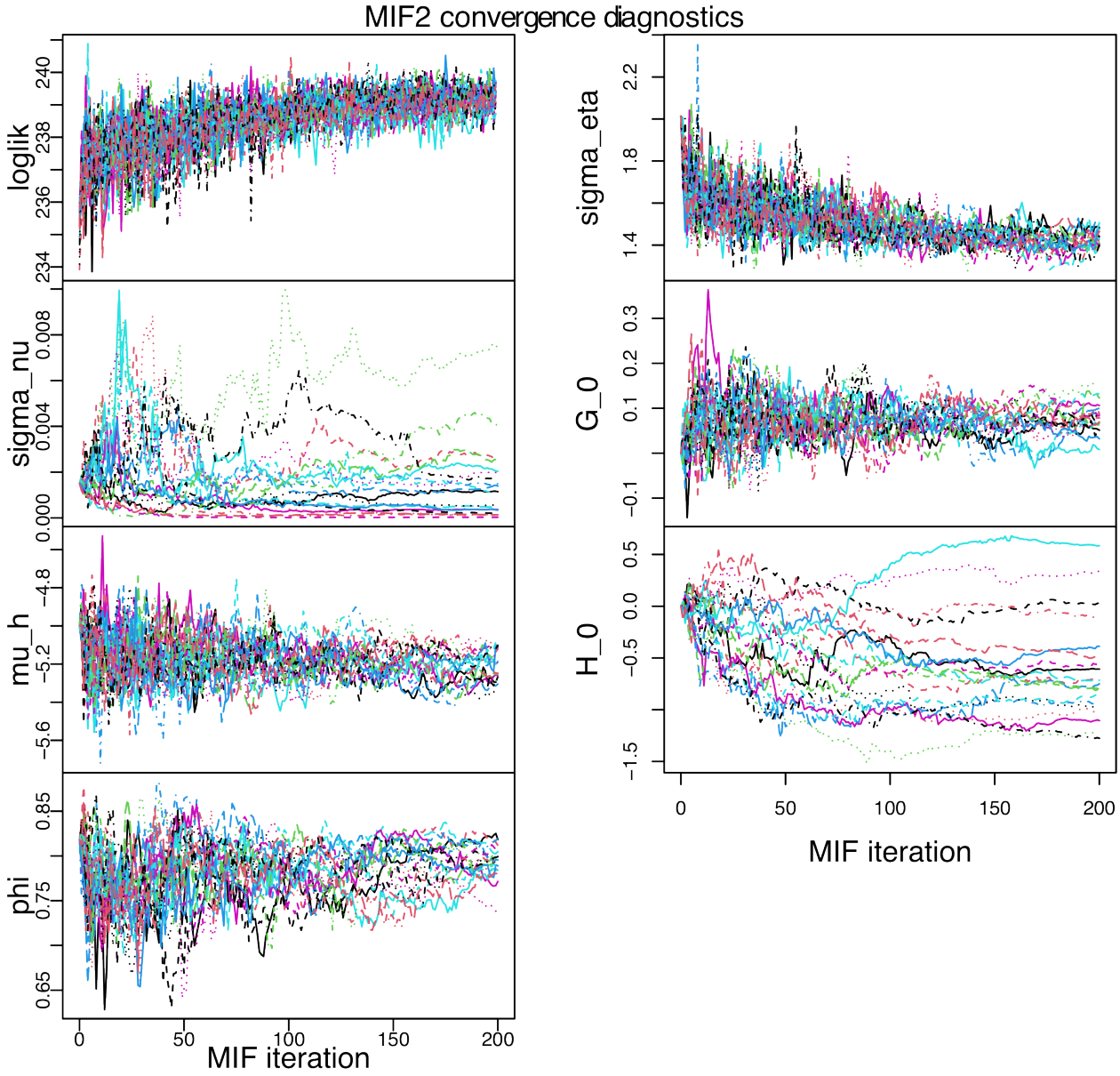

Figure 1: Iterated filtering diagnostic plot

statistical fact worth noticing but not a weakness of the model.

B. The consistently climbing likelihood is promising, but the failure of σ_ν and H_0 to converge needs attention. Additional searching is needed, experimenting with **larger** values of the random walk perturbation standard deviation for these parameters to make sure the parameter space is properly searched.

C. The consistently climbing likelihood is promising, but the failure of σ_ν and H_0 to converge needs attention. Additional searching is needed, experimenting with **smaller** values of the random walk perturbation standard deviation for these parameters to make sure the parameter space is properly searched.

D. The consistently climbing likelihood is promising, but the failure of σ_ν and H_0 to converge needs attention. This indicates weak identifiability which cannot be solved by improving the searching algorithm. Instead, we should change the model, or fix one or more parameters at scientifically plausible values, to resolve the identifiability issue before proceeding.

E. Although the log likelihood seems to be climbing during the search, until the convergence problems with σ_ν and H_0 have been addressed we should not be confident about the successful optimization of the likelihood function or the other parameter estimates.

Solution. A.
All searches are finding parameters with consistent likelihood. The discrepancies of a few log likelihood units put the parameter values within statistical uncertainty according to Wilks's Theorem. Therefore, the spread in the parameter estimates reflects uncertainty about the parameter given the data, rather than a lack of convergence.

That perspective suggests that the goal of the Monte Carlo optimizer is to get close to the MLE, measured by likelihood, rather than to obtain it exactly. Independent Mont Carlo searches can be combined via a profile likelihood to get a more exact point estimate and a confidence interval.

Wide confidence intervals, also called weak identifability, are not necessarily a problem for the scientific investigation. Some parameters may be imprecisely estimable, while others can be obtained more precisely, and part of the analysis is to find which is in each category. It may also be of interest to investigate what extra precision can be obtained on one parameter by making assumptions about the value of another, as in D, but this is not mandatory for proper inference.

Overall, the convergence plots here look good. The plots show that the seaches are all started from a single high likelihood starting point. Now this has been done successfully, a natural next step would be to start some searches from more diverse starting points to look for any global features missed by this local search.

Q10-04.

The iterated filtering convergence diagnostics plot in figure 2 comes from a 2021 student project investigating COVID-19. The calculation used 10^3 particles. What is the best interpretation?

A. Everything seems to be working fine. There is a clear consensus from the different searches concerning the highest likelihood that can be found. Therefore, the search is doing a good job of maximization. Occasional searches get lost, such as the purple line with a low likelihood, but that is not a problem.

B. The searches obtain likelihood values spread over thousands of log units. We would like to see consistent convergence within a few log units. We should use more particles and/or more iterations to achieve this.

C. The searches obtain likelihood values spread over thousands of log units. We would like to see consistent convergence within a few log units. We should compare the best likelihoods obtained with simple statistical models, such as an auto-regressive moving average model, to look for evidence of model misspecification.

D. The searches obtain likelihood values spread over thousands of log units. We would like to see consistent convergence within a few log units. We should look at the effective sample size plot for the best fit we have found yet, to see whether there are problems with the particle filtering.

E. All of B, C, and D.

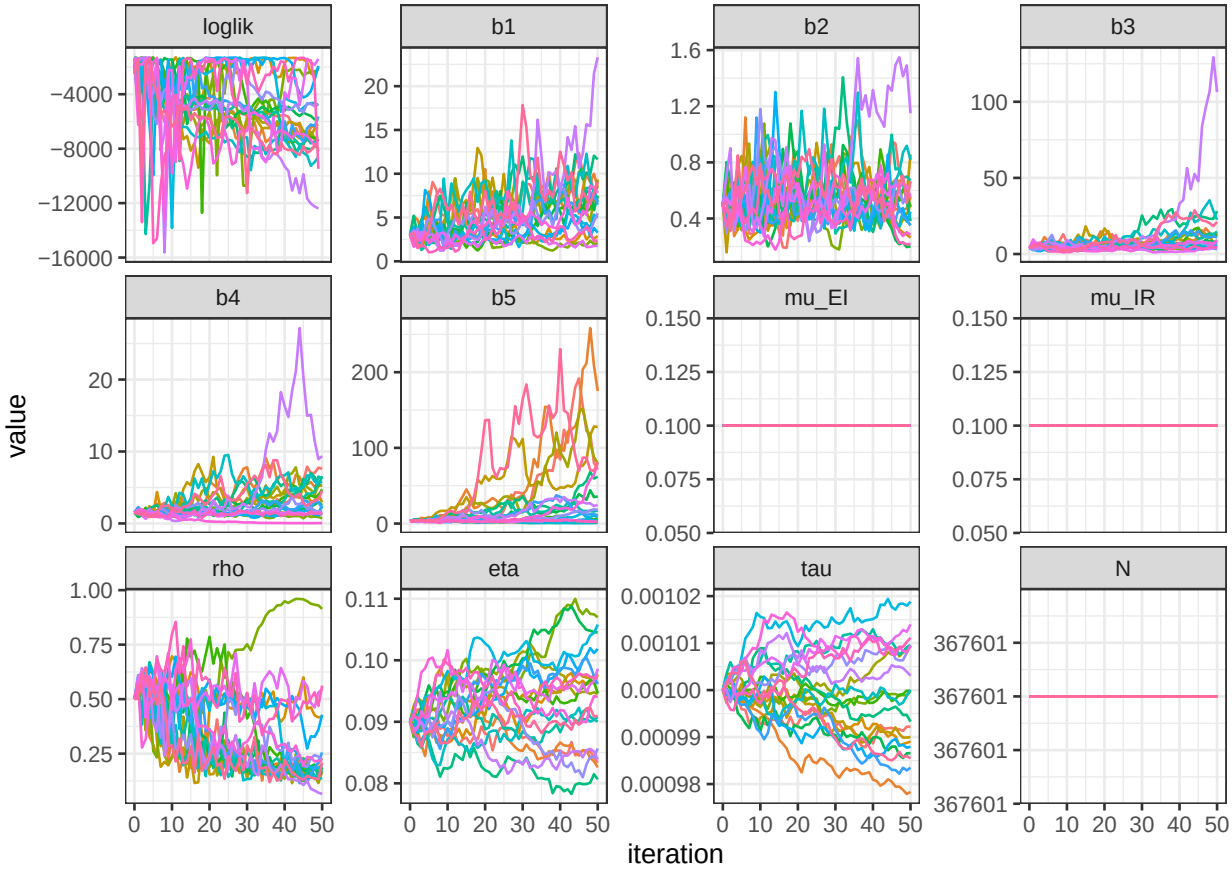

Figure 2: Diagnostic plot for a COVID-19 model

Solution. E.
The wide spread in likelihood, thousands of log units, shown in this convergence plot suggests that the numerics are not working smoothly. The question is, what to do about it?

Once you think your code is debugged, if evidence for poor Monte Carlo convergence remains, it is worth increasing the computational effort to see if that solves the problem. An overnight job on greatlakes could be appropriate. That supports answer B.

Comparing with simple statistical benchmarks is always a good idea. If your model fits more poorly than an ARMA model, or some other simple model appropriate for your data, then it is plausible that the numerical difficulties with the particle filter are due to model misspecification. The diagnostic plots here show some success at global maximization for their model (multiple searches attain a similar likelihood value). Their maximized likelihood was 47 log units lower than ARMA model. This shows that it is worth considering some extra time on model development, supporting answer C. The ARMA benchmark is not given in the question, but you should know that it (or something similar) is useful to calculate.

Identifying time points with low effective sample size can help to identify which parts of the data are problemtic for the model to explain. Thus, D is a standard and useful diagnostic practice. This could help to identify an outlier in the data. Maybe there is an outlier, or maybe there is some other kind of model misspecification; at this point in the investigation we can't tell for sure.

Q4. Data analysis: epidemiological models

Q11-01.

Two models are fitted to case counts on an epidemic. Model 1 is an SIR POMP model with a negative binomial measurement model, and model 2 is a linear regression model estimating a cubic trend. The log likelihoods are $\ell_1 = -2037.91$ and $\ell_2 = -2031.28$ respectively. Which of the following do you agree with most?

A. We should not compare the models using these likelihoods. They correspond to different model structures, so it is an apples-to-oranges comparison.

B. We can compare them, but the difference is in the 4th significant figure, so the likelihoods are statistically indistinguishable.

C. The linear model has a noticeably higher likelihood. Our mechanistic model needs to be updated to beat this benchmark before we can responsibly interpret the fitted model. If a simple regression model has higher likelihood than a more complex mechanistic model, one should prefer the simpler model.

D. The linear model has a noticeably higher likelihood. The mechanistic model is somewhat validated by being not too far behind the simple regression model. We are justified in cautiously interpreting the mechanistic model, while continuing to look for further improvements.

E. The log likelihoods cannot properly be compared as presented, but could be if we used a Gaussian measurement model for the POMP (or a negative binomial generalized linear model instead of least squares for the regression).

Solution. D.
Why not A? Likelihoods of different models for the same data can be compared. Likelihood ratio tests using Wilks's theorem specifically require nested models, but in other contexts (such as AIC and the Neyman-Pearson lemma) the models being compared by likelihood do not need to have any particular relationship.

Why not B? Likelihood ratios have statistical meaning, which corresponds to differences of log likelihoods. The likelihood is a dimensional quantity, whereas the likelihood ratio is dimensionless. The units used correspond to a scientifically arbitrary additive constant to the log likelihood, which disappears after taking differences.

Why not C? If our only goal were to find a predictive model, then (C) could be a reasonable position. Usually, we want to find a model that also has interpretable structure, leading to understanding of the system or

estimating the effect of interventions. A simple regression model cannot do those things, even if it fits a bit better. If the mechanistic model fits much worse than simple alternatives, it is not providing a reasonable explanation of the data, suggesting that there may be important things missing from the model specification.

Quite likely, with some persistence, a mechanistic specification will beat a simple off-the-shelf statistical model.

Q11-02.

A compartment model is first implemented as a system of ordinary differential equations (ODEs). This leads to qualitatively reasonable trajectories, but poor likelihood values. The researchers add stochasticity in an attempt to improve the fit of the model by interpreting the ODEs as rates of a Markov chain. The likelihood, maximized by iterated particle filtering, remains poor compared to ARMA benchmarks. In addition, the effective sample size for the particle filtering is low at many time points despite even using as many as 10^4 particles. Which of the following is the most promising next step?

A. Increase to 10^5 particles, moving the computations to a cluster if necessary.
B. Add noise to one or more rates to allow for overdispersion.
C. Try adding extra features to the model to capture scientific details not present in the original model.
D. Experiment with variations in the iterated filtering procedure; maybe more iterations, or a different cooling schedule.
E. To address the possibility of reporting errors, see if the model fits better when the most problematic data points are removed.

Solution. B.
All the possibilities are worth consideration. However, adding noise in rates to give flexibility in mean-variance relationships is commonly an important part of developing a stochastic model. The simple compartment model interpretation of a ODE as a Markov chain is determined by the rates and therefore does not have free parameters to describe variance. There is some variance inherent in the Markov chain (demographic stochasticity) but additional variability may be needed. It will be hard to investigate the other possibilities if the model has not been given enough stochasticity to explain the variability in the data, so including overdispersion should be an early step. Note that overdispersion can be included in both the process model and the measurement model.

Q11-03. You fit an SEIR model to case reports of an immunizing disease from a city. The resulting confidence interval for the mean latent period is 12–21 days, but clinical evidence points to a latent period averaging about 7 days. Which of the following is the most appropriate response to this discrepancy?

A. The latent period may be confounded with some unmodeled aspect of the system, such as spatial or age structure. The model estimates an effective latent period at the population level, which may not perfectly match what is happening at the scale of individuals.

B. The discrepancy shows that something is substantially wrong with the model. Extra biological detail must be introduced with the goal of bringing the estimated parameter back in line with the known biology of the system.

C. The discrepancy is problematic, but fortunately can readily be fixed. Since we know the clinical value of this parameter with reasonable accuracy, we should simply use this value in the model rather than estimating it.

D. If the model fits the data statistically better than any known alternative model, then we have to take the estimated parameter at face value. It is certainly possible that the estimates in the literature correspond to some different population, or different strain, or have some other measurement bias such as corresponding to severe cases resulting in hospitalization. The discrepancy does not show that our model was wrong.

E. This discrepancy suggests that we should take advantage of both C and D above by putting a Bayesian prior on the latent period. By quantifying the degree of our skepticism about the previously established clinical value of 7 days, we can optimally combine that uncertainty with the evidence from this dataset.

Solution. A.
Transferring parameter estimates between scales is hard. An example is the difficulty of reconciling micro and macro economics. It is generally not possible to guarantee that a parameter means exactly the same thing in models at different scales. (A) acknowledges this. The other answers, in various ways, assume that there should be a single parameter value that describes the system at all scales. There is some merit also to (D), since it is reasonable to try to gain biological understanding by investigating why the fitted model is successful at explaining the data. However, this is an observational study and so we should be cautious of making a causal interpretation of models fitted to data due to the possiblity of confounding. Only (A) addresses this concern.

Q5. Data analysis: financial models

Q12-01.

A generalized autoregressive conditional heteroskedasticity (GARCH) model has $Y_n = \sigma_n Z_n$ where $Z_n \sim$ i.i.d.$N(0,1)$ and

$$\sigma_n^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i Y_{n-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{n-j}^2.$$

For data $y^*_{1:N}$, residuals may be defined by $r_n = Y_n/\hat{\sigma}_n$ where $\hat{\sigma}_n$ is an estimate of σ_n. Suppose that we fit a GARCH model to the log-returns of a financial time series, and we find that the sample ACF of $r_{1:N}$ is consistent with white noise (e.g., 531W24 final project #7). What is the best inference from the residual ACF about the success of the GARCH model for these data?

A. This supports the use of GARCH over ARMA. That is not especially surprising, since it is true for essentially all financial time series, but it is good to check.
B. A fitted ARMA model is also anticipated to have a residual ACF consistent with white noise. The problem with the ARMA model for financial data is not residual autocorrelation.
C. We should also make a normal quantile plot of the residuals. If the residuals are approximately normal then the ACF plot becomes more trustworthy as a test for lack of correlation. If the residuals are far from normal, we should not draw conclusions from the sample ACF.
D. GARCH aims to fix the problem of conditional heteroskedasticity in financial data that ARMA cannot explain. However, fixing this might break the negligible autocorrelation that is critical for the efficient market hypothesis. It is good to see that we can fix conditional heteroskedasticity while remaining compatible with the efficient market hypothesis.

Solution. B.
It would be surprising if substantial sample autocorrelation appeared in the residuals of a GARCH model, at least for a highly traded financial instrument. This would violate the efficient market hypothesis. But that observation is just as true for an i.i.d. white noise model. The reason to prefer GARCH over an i.i.d. white noise model is to explain the conditional heteroskedasticity, but the ACF does not reveal whether or not that is successful.

Q12-02.

A generalized autoregressive conditional heteroskedasticity (GARCH) model has $Y_n = \sigma_n Z_n$ where $Z_n \sim$ i.i.d.$N(0,1)$ and $\sigma_n^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i Y_{n-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{n-j}^2$. There are many extensions to GARCH implemented by various R packages. When comparing models by likelihood or AIC, care is required since packages do not always use standard definitions. What is the most reasonable interpretation of this table?

```
for (i in 1:p) {
  for (j in 1:q) {
    fit_garch <- tseries::garch(log_returns, order = c(i, j))
    garch_table[i, j] <- tseries:::logLik.garch(fit_garch)
  }
}
```

	q1	q2	q3	q4
p1	2646.277	2642.919	2620.280	2616.151
p2	2644.417	2625.417	2622.460	2616.427
p3	2641.804	2637.538	2625.953	2625.740
p4	2639.728	2629.869	2629.969	2628.345

A. The positive values of the log-likelihood are implausible. Perhaps the software actually reports the negative log-likelihood since many optimizers are designed to minimize rather than maximize.
B. The models are nested and so a larger model should mathematically have a larger likelihood. In this table, the larger model usually has lower likelihood, so optimization is problematic.
C. This table would make more sense if `logLik` in fact returns an AIC value. The preferred model is $(p, q) = (1, 4)$.
D. The preferred model is $(p, q) = (1, 1)$ since it is both the simplest model and the one with the highest log-likelihood.
E. `tseries::garch` produces something that is not the likelihood of $y_{1:N}$ or the AIC, and so we cannot readily compare it between models.

Solution. E.

```
?tseries::logLik.garch
```

reveals that

> ‘logLik’ returns the log-likelihood value of the GARCH(p, q) model
> represented by ‘object’ evaluated at the estimated coefficients.
> It is assumed that first max(p, q) values are fixed.

Therefore, the log-likelihood for fiting GARCH(p,q) corresponds only to $y^*_{(\max(p,q)+1):N}$. The violations of nesting occur because different amounts of data are used for differeng values of $\max(p, q)$. Therefore, we cannot easily compare likelihoods or AIC values.

Q12-03.

The Heston model for volatility, V_n, is a stochastic volatility (SV) model with

$$V_n = (1 - \phi)\theta + \phi V_{n-1} + \sqrt{V_{n-1}}\,\omega_n,$$

for $\omega_n \sim N[0, \sigma^2_\omega]$. The log return is $Y_n \sim N[0, V_n]$, conditional on V_n. A previous 531 project (W22, #14) fitted the Heston model to investment in Ethereum, a crypto currency. They obtained a log-likelihood of 34975.3, compared to 28587.4 for GARCH and 28977 for the SV model with leverage presented in class. Their iterated filtering convergence diagnostics are shown in figure 3. What is the best conclusion from this information?

A. The high likelihood shows this is a promising model despite the convergence problems identified in the figure. Attention to the diagnostics may lead to additional improvements.

B. The most important diagnostic feature is the observation that the log-likelihood trace plot peaks and then declines. From the y-axis scale we see the decline is of order 1000 log units. This is evidence of substantial model misspecification which should be addressed.

C. The most important dignostic feature is that the `theta` traces all drop quickly to zero. Since that is not a scientifically plausible value for the parameter, we can deduce that the model is unsuccessful despite its high likelihood.

D. The most important diagnostic feature is that `phi` is close to zero and well identified. This shows that the volatility is close to constant, and is supported by the high likelihood.

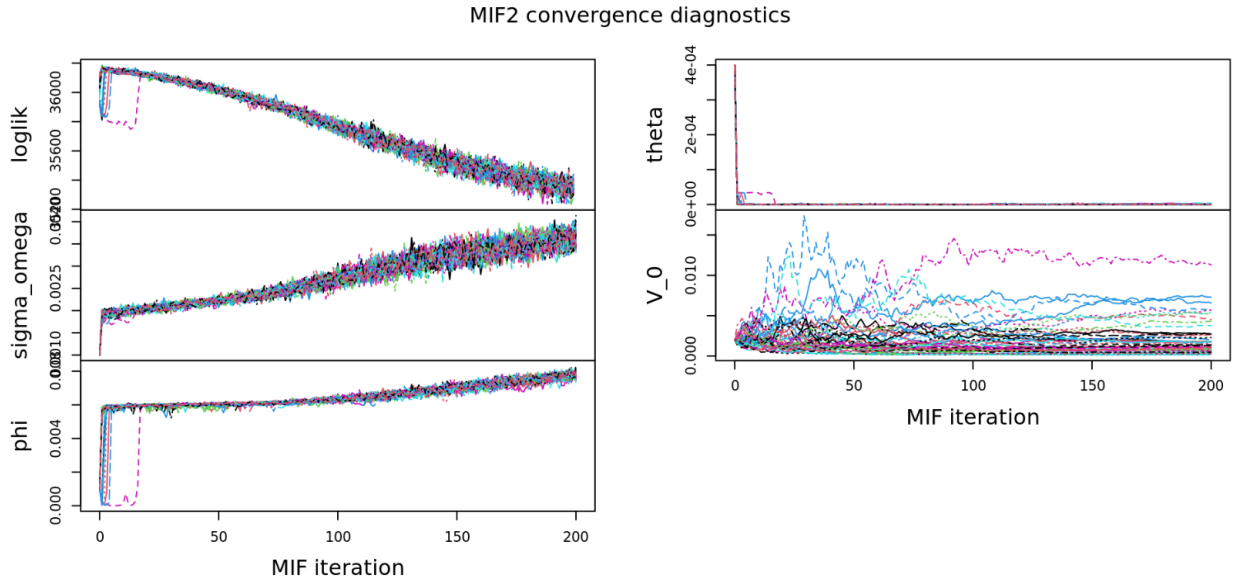

Figure 3: Diagnostic plot for fitting the Heston model

E. The decreasing likelihood and other convergence diagnostics problems show there is a problem with the model. Likely, there is a bug and the high likelihood obtained is simply an error.

Solution. A.
The structure of the particle filter makes it hard to obtain an artificially high likelihood by cheating. Mathematically, the best expected log-likelihood is obtained by a one-step forecast distribution matching the true prediction distribution, assuming the model is correct. (This is just another way of stating the property that the expected log-likelihood is highest under the true model.)

If `dmeasure` is not in fact a density then artificially high likelihoods are possible, but in most models (including this one) the measurement model is a call to a basic R function known to be a density (i.e., integrating to 1).

Inspection of the source code, available online, reveals that the authors made a mistake in implementing `rprocess`. Specifically, the `rprocess` line

```
V = theta*(1 - phi) + phi*sqrt(V) + sqrt(V)*omega;
```

should be

```
V = theta*(1 - phi) + phi*V + sqrt(V)*omega;
```

Thus, their model is not exactly the model they thought they were implementing, leading to incorrect interpretations of their results. Nevertheless, this error turns out to give rise to a model which fits the data very well. This happy accident suggests that a key to modeling the data may be to use a longer-tailed distribution than normal for the returns.

Q6. Computing with POMP models

Q13-01.

Suppose you obtain the following error message when you build your pomp model using C snippets.

```
##
## Error: in 'simulate': error in building shared-object library from C snippets: in 'Cbuilder':
##  compilation error: cannot compile shared-object library
```

```
##  '/tmp/RtmpFkkeCQ/24104/pomp_4fc43714a7a9ebddf896bbc51635d211.so': status = 1
## compiler messages:
## gcc -I"/usr/local/apps/R/ubuntu_20.04/4.2.1/lib64/R/include" -DNDEBUG
##  -I'/home/kingaa/R/x86_64-pc-linux-gnu-library/4.2/pomp/include' -I'/home/kingaa/teach/sbied'
##   -I/usr/local/include   -fpic  -g -O2 -Wall -pedantic -c
## /tmp/RtmpFkkeCQ/24104/pomp_4fc43714a7a9ebddf896bbc51635d211.c
## -o /tmp/RtmpFkkeCQ/24104/pomp_4fc43714a7a9ebddf896bbc51635d211.o
## In file included from /home/kingaa/R/x86_64-pc-linux-gnu-library/4.2/pomp/include/pomp.h:9,
##                  from /tmp/RtmpFkkeCQ/24104/pomp_4fc43714a7a9ebddf896bbc51635d211.c:5:
## /tmp/RtmpFkkeCQ/24104/pomp_4fc43714a7a9ebddf896bbc51635d211.c: In function '__pomp_rmeasure':
## /usr/local/apps/R/ubuntu_20.04/4.2.1/lib64/R/include/Rmath.h:333:16: error:
##  too many arguments to function 'Rf_rnorm
## In addition: Warning message:
## In system2(command = R.home("bin/R"), args = c("CMD", "SHLIB", "-c",  :
##   running command 'PKG_CPPFLAGS="-I'/home/kingaa/R/x86_64-pc-linux-gnu-library/4.2/pomp/include'
##  -I'/home/kingaa/teach/sbied'" '/usr/local/apps/R/ubuntu_20.04/4.2.1/lib64/R/bin/R' CMD SHLIB -c
##  -o /tmp/RtmpFkkeCQ/24104/pomp_4fc43714a7a9ebddf896bbc51635d211.so
##  /tmp/RtmpFkkeCQ/24104/pomp_4fc43714a7a9ebddf896bbc51635d211.c 2>&1' had status 1
```

Which of the following is a plausible cause for this error?

A. Using R syntax within a C function that has the same name as an R function.
B. A parameter is missing from the `paramnames` argument to `pomp`.
C. Indexing past the end of an array because C labels indices starting at 0.
D. Using `beta` as a parameter name when it is a declared C function.
E. A missing semicolon at the end of a line.

Solution. A.
The code producing the error is below. Within C snippets, the C versions of R distribution functions are available but they have slightly different syntax from their more familiar R children. A complete reference guide to R's C interface is available as part of R's documentation. In particular, the C form of R's distribution functions is useful for writing C snippets.

```
sir4 <- simulate(
  sir1,
  statenames=c("S","I","R","cases","W"),
  paramnames=c(
    "gamma","mu","iota",
    "beta1","beta_sd","pop","rho",
    "S_0","I_0","R_0"
  ),
  rmeasure=Csnippet("
    double mean, sd;
    double rep;
    mean = cases*rho;
    sd = sqrt(cases*rho*(1-rho));
    rep = nearbyint(rnorm(1,mean,sd));
    reports = (rep > 0) ? rep : 0;"
  )
)
```

Q13-02. Suppose you obtain the following error message when you build your pomp model using C snippets.

```
##
## Error: error in building shared-object library from C snippets: in 'Cbuilder': compilation error:
##  cannot compile shared-object library
##  '/tmp/RtmpFkkeCQ/24104/pomp_068eedfcaf62b1e391363bbdd99fbe8c.so': status = 1
```

```
## compiler messages:
## gcc -I"/usr/local/apps/R/ubuntu_20.04/4.2.1/lib64/R/include" -DNDEBUG
## -I'/home/kingaa/R/x86_64-pc-linux-gnu-library/4.2/pomp/include' -I'/home/kingaa/teach/sbied'
##   -I/usr/local/include   -fpic  -g -O2 -Wall -pedantic
## -c /tmp/RtmpFkkeCQ/24104/pomp_068eedfcaf62b1e391363bbdd99fbe8c.c
## -o /tmp/RtmpFkkeCQ/24104/pomp_068eedfcaf62b1e391363bbdd99fbe8c.o
## /tmp/RtmpFkkeCQ/24104/pomp_068eedfcaf62b1e391363bbdd99fbe8c.c:
## In function '__pomp_rinit':
## /tmp/RtmpFkkeCQ/24104/pomp_068eedfcaf62b1e391363bbdd99fbe8c.c:38:13:
##  error: called object is not a function or function pointer
##    38 |     cases = 0
##       |             ^
## make: *** [/usr/local/apps/R/ubuntu_20.04/4.2.1/lib64/R/etc/Makeconf:168:
##  /tmp/RtmpFkkeCQ/24104/pomp_068eedfcaf62b1e391363bbdd99fbe8c.o] Error 1
## In addition: Warning message:
## In system2(command = R.home("bin/R"), args = c("CMD", "SHLIB", "-c",  :
##   running command 'PKG_CPPFLAGS="-I'/home/kingaa/R/x86_64-pc-linux-gnu-library/4.2/pomp/include'
##  -I'/home/kingaa/teach/sbied'" '/usr/local/apps/R/ubuntu_20.04/4.2.1/lib64/R/bin/R' CMD SHLIB -c
##  -o /tmp/RtmpFkkeCQ/24104/pomp_068eedfcaf62b1e391363bbdd99fbe8c.so
##  /tmp/RtmpFkkeCQ/24104/pomp_068eedfcaf62b1e391363bbdd99fbe8c.c 2>&1' had status 1
```

Which of the following is a plausible cause for this error?

A. Using R syntax within a C function that has the same name as an R function.
B. A parameter is missing from the `paramnames` argument to `pomp`.
C. Indexing past the end of an array because C labels indices starting at 0.
D. Using `beta` as a parameter name when it is a declared C function.
E. A missing semicolon at the end of a line.

Solution. E.
The error message was produced by the code below. `pomp` passes on the C compiler error message for you to inspect. Note the missing semicolon in the next-to-last line.

```
sir1 <- sir()
sir2 <- pomp(
  sir1,
  statenames=c("S","I","R","cases","W"),
  paramnames=c(
    "gamma","mu","iota",
    "beta1","beta_sd","pop","rho",
    "S_0","I_0","R_0"
  ),
  rinit=Csnippet("
  double m = pop/(S_0+I_0+R_0);
  S = nearbyint(m*S_0);
  I = nearbyint(m*I_0);
  R = nearbyint(m*R_0);
  cases = 0
  W = 0;"
  )
)
```

Q13-03.

A useful way to check statistical methodology is to apply an inference method to a collection of simulated datasets from the fitted model with the estimated parameter values (say, the maximum likelihood estimate, MLE). This is sometimes called a "parametric bootstrap". Suppose that we carry out this check for a POMP

data analysis, using plug-and-play inference methodology such as iterated filtering, and we find that the re-estimated parameters from inference on the simulated data are close to the MLE. What can we infer about the correctness of our inference.

A. This is a strong check that both the model and the methodology are correctly implemented. Except for some rare special cases, an error in either one of these will lead the check to fail.

B. This checks the implementation of the inference methodology but not the model. Even if the model is implemented wrongly, the check will still show us whether the inference methodology is correct.

C. This checks the implementation of the model but not the inference methodology. As long as the model is implemented correctly, any reasonable inference methodology should pass the check successfully.

D. This is not a strong check of either the model or the methodology. It shows self-consistency but that is different from showing accuracy.

Solution. B.

This is a useful property to bear in mind when debugging statistical analysis carried out using plug-and-play methodology. By definition, the inference methodology defines the model via a simulator, and presumably the same simulator is used for inference as for the simulation used to test the inference. Thus, the parametric bootstrap exercise tests the inference methodology but not the correctness of the model implementation; errors in the latter will apply in the same way to both the simulation and the inference, so cannot show up as a mismatch between inferred parameters and re-estimated parameters.

Errors in `rprocess` for a POMP model are hard to debug for this reason. Best practice is to present the Csnippet right next to the math representation, and use the same notation for both, so that the visual match is evident.

It is a good idea to carry out a parametric bootstrap despite this limitation.
